Final Report to

RPSEA
Research Partnership to Secure Energy for America

SeTES, a Self-Teaching Expert System for the analysis, design and prediction of gas production from unconventional resources

07122-23.FINAL

November 28, 2011

PI: George J. Moridis (gjmoridis@lbl.gov)

Co-PIs: Matthew T. Reagan (LBNL), Thomas A. Blasingame (TAMU), Michael Nikolaou (UH)

Additional Authors: Heidi A. Kuzma, Ralph A. Santos, Katie L. Boyle

Lawrence Berkeley National Laboratory

1 Cyclotron Rd., Berkeley, CA 94720

(510) 486-4746

LEGAL NOTICE

This report was prepared by Dr. George Moridis as an account of work sponsored by the Research Partnership to Secure Energy for America, RPSEA. Neither RPSEA members of RPSEA, the National Energy Technology Laboratory, the U.S. Department of Energy, nor any person acting on behalf of any of the entities:

a. MAKES ANY WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED WITH RESPECT TO ACCURACY, COMPLETENESS, OR USEFULNESS OF THE INFORMATION CONTAINED IN THIS DOCUMENT, OR THAT THE USE OF ANY INFORMATION, APPARATUS, METHOD, OR PROCESS DISCLOSED IN THIS DOCUMENT MAY NOT INFRINGE PRIVATELY OWNED RIGHTS, OR

b. ASSUMES ANY LIABILITY WITH RESPECT TO THE USE OF, OR FOR ANY AND ALL DAMAGES RESULTING FROM THE USE OF, ANY INFORMATION, APPARATUS, METHOD, OR PROCESS DISCLOSED IN THIS DOCUMENT.

THIS IS A FINAL REPORT. THE DATA, CALCULATIONS, INFORMATION, CONCLUSIONS, AND/OR RECOMMENDATIONS REPORTED HEREIN ARE THE PROPERTY OF THE U.S. DEPARTMENT OF ENERGY.

REFERENCE TO TRADE NAMES OR SPECIFIC COMMERCIAL PRODUCTS, COMMODITIES, OR SERVICES IN THIS REPORT DOES NOT REPRESENT OR CONSTIITUTE AND ENDORSEMENT, RECOMMENDATION, OR FAVORING BY RPSEA OR ITS CONTRACTORS OF THE SPECIFIC COMMERCIAL PRODUCT, COMMODITY, OR SERVICE.

[blank page]

Signed:

George J. Moridis

December 1, 2011

Date

Abstract

SeTES is a self-teaching expert system that (a) can incorporate evolving databases involving any type and amount of relevant data (geological, geophysical, geomechanical, stimulation, petrophysical, reservoir, production, etc.) originating from unconventional gas reservoirs, i.e., tight sands, shale or coalbeds, (b) can continuously update its built-in 'public' database and refine the its underlying decision-making metrics and process, (c) can make recommendations about well stimulation, well location, orientation, design and operation, (d) offers predictions of the performance of proposed wells (and quantitative estimates of the corresponding uncertainty), and (e) permits the analysis of data from installed wells for parameter estimation and continuous expansion of its database. Thus, SeTES integrates and processes any available information from multiple and diverse sources on a continuous basis to make recommendations and support decision making at multiple time-scales, while expanding its internal database and explicitly addressing uncertainty. It uses three types of data: public data, that have been made available by various contributors, semi-public data, which conceal some identifying aspects but are available to compute important statistics, and a user's private data, which can be protected and used for more targeted design and decision making. SeTES can be a vital and easy-to-use tool in gas production from unconventional gas resources. It is expected to result in a significant increase in both reserve estimates and production by providing a technology that will increase efficiency and reduce the uncertainties associated with such gas reservoirs.

EXECUTIVE SUMMARY

Project Manager: George Moridis, Lawrence Berkeley National Laboratory (LBNL)

Principal Investigators: George Moridis and Matthew Reagan, LBNL, James Rector, LBNL and University of California at Berkeley (UCB), Thomas Blasingame (Texas A&M University - TAMU), Michael Nikolaou (University of Houston - UH)

Project Title: A Self-Teaching Expert System for the Analysis, Design and Prediction of Gas Production from Unconventional Gas Resources

OBJECTIVES: Using a multi-disciplinary approach, to develop a self-teaching expert system that (a) can incorporate evolving databases involving any type and amount of relevant data (geological, geophysical, geomechanical, stimulation, petrophysical, reservoir, production, etc.) originating from unconventional gas reservoirs, i.e., tight sands, shale or coalbeds, (b) can continuously update its built-in 'public' database and refine the its underlying decision-making metrics and process, (c) can make recommendations about well stimulation, well location, orientation, design and operation, (d) offers predictions of the performance of proposed wells (and quantitative estimates of the corresponding uncertainty), and (e) permits the analysis of data from installed wells for parameter estimation and continuous expansion of its database.

DELIVERABLE: The deliverable of this project is an alpha release of a self-teaching expert system that can be a vital tool in the attempt to increase reserves and successfully produce gas from shale formations, and to increase production from already producing systems. The final product is not just the development of an abstract approach or methodology, but a computer program that is easily installed and executed on a wide variety of computational platforms. To fully realize the benefits of the self-teaching expert system, the approach is storage at a central location and access through a Web-based application. Note that the data that are entered into the database are treated as confidential, with the user not knowing their origin without the explicit consent of the data owners. Although the geographical location associated with the data may be disclosed, the data provenance and ownership will not. Thus, the user benefits from the data availability to design more productive production systems without compromising confidential information belonging to the entities that provided the data.

POTENTIAL IMPACT: Successful development of the proposed self-teaching expert system is expected to result in a significant increase in both reserves and production by providing a technology that will significantly reduce the uncertainties associated with such systems, thus bringing previously inaccessible energy resource to production.

MAJOR INDUSTRIAL PARTNERS: Anadarko, BGI.

Table of Contents

REPORT DETAILS

1. Description of the problem and proposed solution

Unconventional gas resources (UGRs) are best described as gas accumulations that are hard to characterize and commercially produce by common exploration and production technologies. These resources are typically located in very tight, heterogeneous, extremely complex, and often poorly understood geologic systems, often easy to find but difficult to produce. Thus, while it is not difficult to find extensive UGRs in many basins, it is very difficult to determine their flow properties from petrophysical well surveys and to design effective completion procedures. Furthermore, because of their very low permeability, establishing gas flow at commercial rates requires costly well stimulation operations. These considerations (low permeability and complex geology) are responsible for the high risk factors and unpredictable results often associated with UGR exploration and development projects, and hamper industry investment in these resources despite their potentially vast magnitude.

The problem of tight gas production has inherent and multifaceted uncertainty (e.g., in reservoir properties or gas prices) coupled with overwhelming complexity in making decisions at all phases of gas field development. While the problem of making decisions for complex systems in an environment of high uncertainty appears in many engineered systems (e.g. chemical plants, managed ecosystems, transportation systems) gas production from very tight reservoirs has its own intricacies, which make the application of standard methods and tools far from trivial, if at all feasible. Nevertheless, recent work [*Saputelli* et al., 2005] has demonstrated that a multi-scale approach to the overall problem of field development can tame complexity, mitigate uncertainty, and coordinate information processing from a multitude of sources and varied formats. This approach involves a rational development of tools and business processes that are manageable, do not have internal conflicts, complement each other, and contribute towards optimizing an otherwise difficult to manage system. This multi-level methodology has been applied widely to the oil refining industry, for which a multitude of tools at different levels have been developed over the last three decades with impressive results.

Concerted development of the same methodology has advanced far less in the oil and gas production industry, although considerable efforts have appeared in recent years, some of which have produced positive results. Terms such as “intelligent fields” or similar have been used to describe the use of modern computing and communication tools for the development and operation of oil and gas fields. However, such tools are little developed for UGR. A variety of tools are used for a multitude of tasks associated with each level of the hierarchy in UGR development (e.g., seismic, reservoir simulation, core analysis, etc.). Yet, use of such tools is fragmented and unwieldy, teams using such tools are often compartmentalized, and much potentially useful information remains unused simply because of ineffective communications or because the right tools or processes have not been developed. As a result, less than optimal decisions may be reached, resulting in the "hit-or-miss" approach that characterizes practically all aspects of UGR exploration and resource development.

Consequently, there is a clear need, as well as an opportunity, to develop field development and operation tools that are a) modular, b) integrated in the form of manageable communicating modules that serve an overarching objective (maximize production, recovery, net present value or other related objective), and c) easy to use and maintain. To that end, we developed a Self-Teaching Expert System (SeTES) for the analysis, design, and prediction of gas production from unconventional gas resources. Note that we use the term Expert System (ES) to denote an *integrated collection of tools rather* than a specific piece of artificial intelligence software.

SeTES is intended as public-domain software. It is currently available as an Alpha-2 release, and can be provided to interested parties upon request. The most innovative aspects of SeTES are its handling of all relevant data types (geological data, geophysical data, petrophysical and reservoir data, well completion data, production data, etc.), its ability to integrate new data directly into the system, and its ability to continuously update and refine the decision metrics for recommendations and predictions. To our knowledge, no operator applies any type of formal ES, let alone self-teaching ones, and SeTES is the first such tool ever developed.

2. Original Objectives

The original objectives of the project were to develop a self-teaching expert system that could:

1. Incorporate evolving geological, geophysical, fracturing, reservoir and production data obtained from a continuously expanding database of installed wells in unconventional tight gas reservoirs.

2. Continuously update the built-in database and refine the underlining decision-making metrics and processes.

3. Make recommendations about formation fracturing and well stimulation, in addition to well location, orientation, design and operation using the most recently updated metrics and processes.

4. Offer predictions of the performance of proposed wells (and quantitative estimates of the corresponding uncertainty) in the stimulated formation.

5. Permit the analysis of data from installed wells for parameter estimation and continuous expansion of the database of the expert system.

Furthermore, the deliverable of the project was to be not just the development of an abstract methodology, but an Alpha-release of a computer program that could be installed and executed on a wide variety of computational platforms.

A last requirement was decided on early in the project as a consequence of the fact that the system was going to have to be freely available, and that is that the system be built entirely with freeware or open source tools so that there would never be any licensing or copy write issues.

In the following sections, we discuss how each of these objectives became a challenge and how each challenge was ultimately met, including the difficulties encountered, the successes and failures, the adaptation of the original goals, and the solutions that have been discovered but not yet implemented. Further details are given in section 4.2, Technical Implementation.

Typical software projects, particularly large-scale ones, undergo many months of detailed

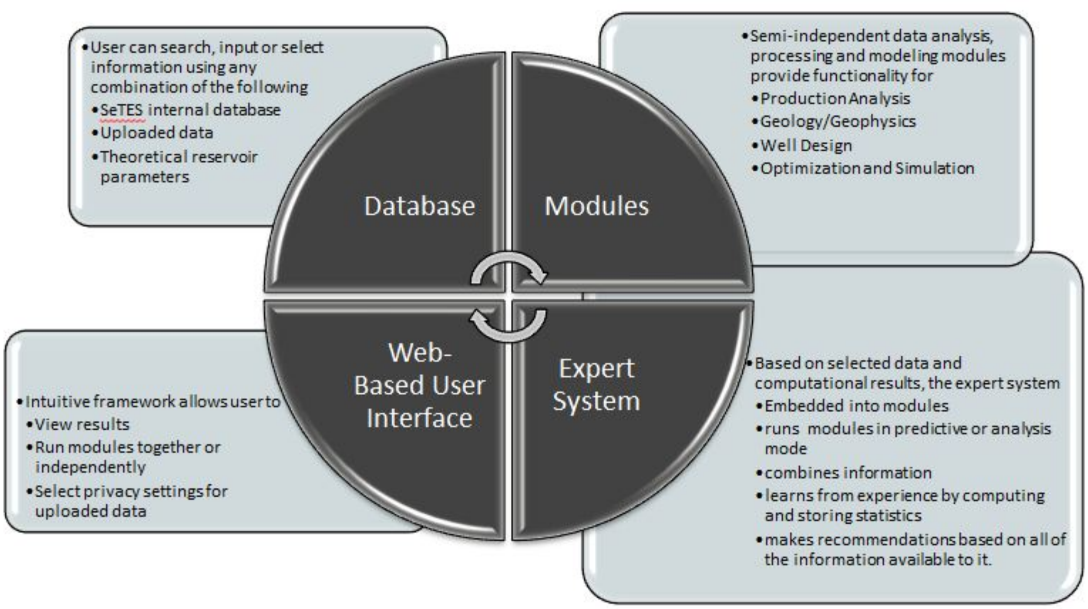

Figure 1: Design of SeTES.

SeTES is composed of a set of semi-independent processing modules, a database and a web-based user interface. The Expert System of SeTES can be thought of as a separate component of the system. In actuality, it is integrated with the modules in the form of built-in intelligence, with the database in as a mechanism for remembering statistics and the user interface as a guided user experience.

planning before they are implemented. This was impossible with SeTES because, although primary objectives were stated at the onset, there was, and could not be, a coherent initial vision of the project for the simple reason that nothing like it had been ever been produced. This meant that a considerable amount of time, especially in the first year of the project, was spent inventing ideas that may or may not have ended up being utilized in the final system. The best of these ideas are also discussed in this report.

3. Objectives Become Challenges

In order to meet all of the objectives of the original proposal as well as other challenges that were brought on during development, we came up with a four-part design for SeTES, which is illustrated in Figure 1. SeTES is composed of:

1. A set of semi-independent computation modules
2. A database
3. A user-friendly website and
4. An Expert System which is disseminated throughout the first three elements as
    a. Intelligence built into the modules
    b. Self-learning stored in the database
    c. Guidance for the user built into the website

3.1 Challenge: Building an Expert System

An expert system is defined loosely in Wikipedia as a "computer program which stands in for the experience of talking to a number of human experts." There is, however, no general consensus as to what the words mean, specifically. Worse still, in the computer learning and Artificial Intelligence communities, the failure of early expert systems to be of much use or interest is often credited with the so-called "A.I. Winter," a period in the 1980's and 1990's when the entire concept of A.I. was largely discredited. As such, many modern computer systems that fit the description of an Expert System are not called by that name.

Early textbooks in A.I. and computer learning often make mention of an early expert system called PROSPECTOR [Duda et al, 1977], a system developed at Stanford in the mid-1970s to discover and evaluate hard-rock mineral systems. Despite successors PROSPECTOR II [McCammon, 1994], and PROSPECTOR III (mentioned in [Agerberg, 1989] but apparently otherwise unpublished), the system remained more of a research project and proof-of-concept than a commonly used exploration tool[1].

[1] Private conversations with exploration scientists who had seen and used PROSPECTOR indicate that although it was considered a triumph in computer science, it wasn't a very useful prospecting tool. A mineral deposit which it is credited with discovering was, in fact, well known at the time and in any case was not important enough to mine.

In the mid 1980s and early 1990s, there were a few short-lived attempts to build very limited systems for Earth science using Graphical Information Systems (GIS) methods, such as MAP-AID [Robinson and Jackson, 1985], ACES [Pfeferkorn et al, 1985], ACKRONYM [Brooks, 1983] and CES [Muller et al, 1986]. Ramani and Gautum [1990], Rao [1991] and Rao et al. [1990] also proposed GIS systems that did not reach public release. A GIS system which emulated PROSPECTOR was proposed by Katz [1991], and a promising Case Based Reasoning (CBR) system for oil field design has been proposed by Kravis and Irrgang [2005]. The latter works by simply matching new data with a database and asking a user for confirmation and has no mechanism for self-learning.

The lack of existing, widely accepted expert systems in general meant that, although the intuitive concept of an expert system might be clear, there existed no concrete model to follow or draw on in this project. By far, the biggest challenge in the SeTES project has been to try to determine what a modern expert system should look like and how to implement it with enough flexibility that it can be updated as new knowledge becomes available.

Early expert systems were typically implemented as a Q&A session in which a user was asked a series of questions, the answers to which were plugged into a hardwired decision-tree structure. The system would make decisions and suggestions based on probabilistic outcomes in the tree. After an early attempt by us to implement such a system for this project, we realized that trying to simulate a conversation was not going to be a fruitful path for SeTES. It would simply confuse and frustrate modern users who are comfortable with computers and expect to be able to navigate through a program using windows, menus, and tabs. Furthermore, we realized that the tools and algorithms that SeTES would use would be useful as stand-alone applications, and that a user who had no interest in the expert system might still want access to it in order to use a particular application.

This led us to the concept of building the system as a set of linked processing modules not unlike the individual sheets and scripts within a spreadsheet. Except that, unlike a spreadsheet, the modules themselves have integrated intelligence. Each module is intended to perform a specific task (such as decline curve analysis or hydro-fracking fluid selection) and is literally designed by a human “expert,” that is, a professional in the petroleum engineering or computer learning field. The modules communicate with each other and with an underlying database by continually asking the equivalent of the question “what has been done in this analysis, and what can I do next?” The modules are designed to run in several different modes depending on what kind of data is available and what other modules have already produced results.

The modular design of the system is key to its future flexibility. New processing capability can be added to the system by adding a new module. Because the intelligence of the system is built into the modules, it means that adding a new module can be done without disrupting the stability and functionality of the system.

3.2 Challenge: Making the System Self-Teaching

SeTES is a self-teaching expert system, which means that, although the system can evolve over time through the addition of new modules, it must also have an internal mechanism for incorporating new experience.

The design of a self-teaching feature has been challenging, not least because the initial dataset made available courtesy of Anadarko Corp is relatively small. Furthermore, in light of an industry-wide concern for the privacy of data, it is not clear that large amounts of data may ever be uploaded into the system—at least not into the public database. Therefore, designing a system that relies on seeing huge amounts of data in order to tailor its outputs to a particular application (such as we see with popular sites such as Google or Facebook), was not only impractical, but might never be fruitful.

In the process of development, we observed that the output of quite a few of the SeTES modules, especially those involved in curve-fitting and optimization, are very sensitive to initial guesses. For example, the decline-curve fitting module will not converge to an estimate of parameters unless it has been giving a reasonable initial guess. Thus, instead of remembering all of data that it has been shown, SeTES is designed to remember statistics on parameters that have been computed previously and to use those statistics as a starting point for each new analysis. In this way, the system gains the capacity to learn with experience and its solutions should improve over time. The self-teaching aspect of the system will become more and more apparent as more data is added to the system over time.

3.3 Challenge: Incorporating Evolving Data

We required that SeTES predictions should improve as more data about a particular well, field, or formation is added to the system. We also wanted the system to be able to start from almost nothing, using just a hint such as the general coordinates of a well or the name of a formation or field. We wanted the predictions to improve as more data, such as production records, geophysical well-logs, completion reports and laboratory data were added.

We solved the problem of sparse data using a system of intelligent defaults. By checking the statistics stored in the database, SeTES often contains information about what the values of, say, reservoir permeability and porosity ought to be for a particular known formation. If it has no reason to believe otherwise, i.e. it has computed nothing else or the user has not entered an override value, then it will use these defaults.

3.4 Challenge: Continuously Updating the Database and Underlying Decision-making Metrics

This challenge is a result of both the self-teaching requirement of the system and the fact that SeTES cannot be static. There must be a mechanism for a user to upload data into the system and for the system to make use of it and combine it with existing data.

The issue of moving data through the system has proven to be complicated, and no doubt will continue to be. It has required that we design a complete database capable of holding a variety of different data types. Furthermore, it required that we design data readers that would be able to parse data from many different formats. Because of the limited timeframe, our Alpha-release routines are limited to parsing the format used by Anadarko's Holly Branch data set. However, we recognize that in the future reading diverse collections of data may become a huge issue in the success of the system.

At the initiation of a user session, data is submitted to the system through a data manager window, which the user can access throughout the system. Before each module runs, it first checks to see what kind of information is available to it. If a module is unable to run, the module will suggest that the user upload or select appropriate data. Since the data manager is always available, the user can add information as desired, re-computing all of the results within the session by simply clicking a restart button.

3.5 Challenge: Making Recommendations Using Updated Metrics and Processes

SeTES updates itself internally by remembering statics on data that it has already seen and parameters that it has computed. It feeds this information back to its modules when they need it to come up with informed initial guesses, as prior distributions, or as intelligent defaults. The system also needs to be flexible such that when entirely new ways of doing things become available, such as new optimization algorithms or updated data processing routines, they can be integrated into the system. This can be done by adding a new module to the system.

3.6 Challenge: Analyzing Data for Parameter Estimation and Database Expansion

SeTES modules can run in one of several modes:

- Analysis mode: modules estimate parameters from data
- Simulation mode: modules use parameters to simulate data
- Optimization mode: modules use parameters and simulations to make recommendations

Many of the SeTES modules run in more than one mode: if data is available, they estimate parameters (such as permeability, fracture length etc.), or if there is no data but the user has entered approximate values for parameters or intelligent defaults exist, they generate new results (such as production curves, pressure fields or estimated ultimate recovery).

The first thing a module does when called is to consult the database for useful information. If it finds data that can be used to estimate parameters, it automatically does so. If not, it might automatically switch into predictive mode and show the user what the data for this particular test ought to look like, using as a first choice parameters that have been entered by the user and, if nothing better is available, default values from the database.

Furthermore, each of the data analysis modules can be run independently. A user may upload data and run a module without using any other part of the expert system, in so desired. A user who only wants to use the Rate/Pressure analysis module, for instance, can log in and do so directly, without uploading or selecting any other type of data. The results from this analysis, however, would be incorporated into the database and be used by optimization modules at a later date if the user gives permission (see privacy choices under section 4.1.2 Data Manager and Data Privacy).

3.7 Challenge: Building a Computer Program Available Online

Early in the project, it was decided that SeTES would be a web-based tool that would run on a server at LBNL and be accessible, publically, over the internet. This requirement added an enormous amount of overhead to the project, because programming for the web is significantly more complicated that programming a stand-alone application with a graphical user interface. However, it is not clear that the system would have developed as it did without this challenging requirement.

SeTES, or at least some of its functionality, could have been developed in a more user-friendly (for the programmer) environment such as MATLAB or EXCEL. If we had done so, we would have seen immediate results. However, the job of translating the tools into higher-level languages and migrating them to a web platform would have been enormous, and the limitations of programming within existing desktop applications may have severely hindered development of many of the new technologies seen in SeTES. Furthermore, since we were immediately faced with web constraints, we were forced to develop applications appropriate for a web platform rather than “fantasy applications” that would have been impossible to implement. As it is, the entire system has had to be completely overhauled twice during its development, moving towards a system with a uniform underlying platform and development framework. However, if we had not developed on a web platform from the onset, problems of integration would have been significantly worse, if not intractable, and we might have abandoned the idea of web deployment altogether.

The fact that SeTES and all of its modules and applications can be run over the internet is one of its greatest strengths, making it much more interesting than a simple expert system because the user has access to a wide variety of processing tools in a straightforward environment.

As far as being a standalone system, SeTES, can, with a little guidance, be run as a private application—a disk image of the system can be downloaded and installed in a private server environment, or the system software can be configured from scratch on any web stack. In fact, this is how the SeTES development team works, using private copies of the systems installed on a variety of laptops so that they can experiment and alter the system without affecting the stable public release or corrupt the public database. New versions and updates are communicated through a github (www.github.com) account that acts as a central, safe, version-controlled repository for new code. Any person interested in becoming a SeTES developer or getting access to the code can request access through the development team. The disk image can also be directly delivered to a customer and deployed on any computer using commercially available virtualization software. However, since the most recent version of the system is easily accessible over the internet, the only reasons to set up a private system would be either to add new, private features and modules to the system or to run in an environment of absolute data security.

3.8 Challenge: Using Only Open Source and Free Software

Most of the early processing modules were developed in third-party commercial application environments (such as EXCEL or MATLAB). Not only is it difficult to integrate this kind of software in a website, but running it would require that the website and possible each individual user acquire all of the appropriate licenses as well. This would have been unacceptable.

After considerable discussion, we chose to use the popular freeware scripting languages Python and PHP as the core languages of the system. PHP is a standard tool for web development. In SeTES, it handles most aspects of the website and communication with the database. The modules themselves are implemented in Python, a more robust programming language with mathematical and scientific capabilities.

The choice of Python was less obvious than the decision to use PHP. The advantages of using Python are that it is a well-developed, free language that integrates well with PHP and contains a number of freeware packages for computation and plotting. It is object-oriented, which is a double-edged sword, making it easy to modularize and combine with PHP but complicating the implementation of numerical algorithms. By far the biggest drawback of the language is that, unlike MATLAB, Fortran and EXCEL, Python is not at all commonly used in petroleum engineering and geophysics, which means that the core LBNL programming team had to re-write or otherwise encapsulate much of what was developed by the subject-matter experts on the team. Another issue with Python is that it has gone through many incompatible versions, so finding a version that was stable and included the features we wanted was a challenge. Upgrades are nearly impossible.

Early in development, we experimented with a freeware MATLAB emulator called Octave, however as Octave and MATLAB source code is not truly interchangeable, it was soon clear that it would require only somewhat more effort to completely rewrite the modules in Python. On a positive note, however, our Python implementation has left us with a package of routines that, with only a small investment of time, could be easily distributed through a freeware website such as www.SourceForge.net for general use.

At several times during development, we second-guessed our choice to use freeware and develop directly as a web application using Python, and reconsidered moving development to a simpler environment. However, in retrospect, despite the difficulty, we made the correct choice, because we now have a system that may have less functionality and fewer modules that it otherwise would have, but which is fully implemented, meaning that it can be deployed, improved and added to directly without having to go through a second development period. Moreso, SeTES is now usable online through any modern web browser (including via laptops and handheld devices), and it not "trapped" within a particular proprietary application or restricted to a single operating system.

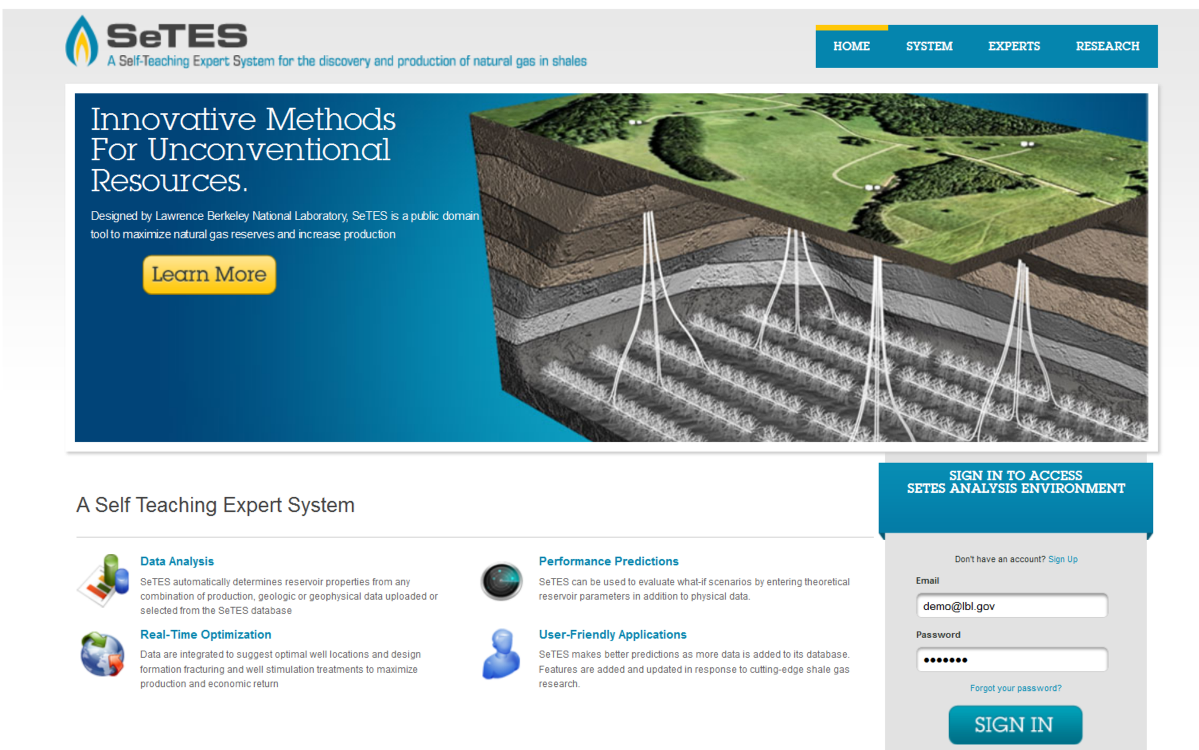

Figure 2: The SeTES Homepage.

3.9 Challenge: Delivering a Website

Because SeTES is a website, it requires a homepage, as well as several other pages outside of the analysis environment. And although the SeTES team was adept at designing algorithms, writing code, interpreting shale gas data, and writing academic papers, designing a useful and attractive graphical webpage was not one of our core competencies. In the final months of the project, it became clear that if the SeTES website was going to be effective, an attractive homepage would have to be put together by professionals with graphic design experience. We outsourced the graphic design to New Leaders (www.NewLeaders.com) and hired a technical editor to manage the text, both through East Donner Research LLC (www.EastDonnerResearch.com). The results are significantly more aesthetically pleasing and coherent than what we had attempted to produce on our own. As of this phase of development, the SeTES website has a Homepage, a Sign-In to the Analysis Environment, and four other secondary pages, including one that contains all of the research generated by the project available by web link.

4. The SeTES website and user Experience

The SeTES website is the most comprehensive deliverable for the SeTES project. It is the first site of its kind to offer a database, a set of working analysis tools, a demonstration environment, and a research repository for shale gas drilling which is free to the public. All of the research produced as a result of the SeTES project is available on the website, along with the algorithms driving the modules, and the website will continue to publish research and news related to shale gas that may be of interest to its readers. It will also continue to grow by adding to its database of well informatics, thereby improving its predictive capacity, and by making new analysis modules available as they are written. Finally, the website may expand to include an environmental component, addressing issues that are of increasing concern to the public and to the public perception of shale gas drilling. Figure 2 shows a screenshot of the SeTES homepage.

The SeTES website aims to serve a variety of users and communities, including:

- Professionals (petroleum engineers, producers, leaseholders, investors)
- Academics (researchers, students)
- The General Public (energy users interested in shale gas)

Making the SeTES website user-friendly for non-specialists has been a goal of the project and will continue to be important as the website grows. Text describing shale gas research is written in accessible language, as are instructions for using the Analysis Environment and understanding the output.

The body of the Homepage explains the primary functional aspects of the website with four icons and their descriptions, namely:

1. **Data Analysis** (SeTES automatically determines reservoir properties from any combination of production, geologic or geophysical data uploaded or selected from the SeTES database. SeTES estimates uncertainty, estimates petrophysical parameters and combines evidence.)
2. **Performance Predictions** (SeTES can be used to evaluate what-if scenarios by entering theoretical reservoir parameters in addition to physical data.)
3. **Real-Time Optimization** (Data are integrated to suggest optimal well locations and design formation fracturing and well stimulation treatments to maximize production and economic return.)
4. **User-Friendly Applications** (SeTES makes better predictions as more data is added to its database. Features are added and updated in response to cutting-edge shale gas research. Applications run in a self-explanatory, interactive, transparent environment.)

There is an additional box on the Homepage for **Recent Activity**, which list additions to the website, including new research papers, new modules, new data, or any other news

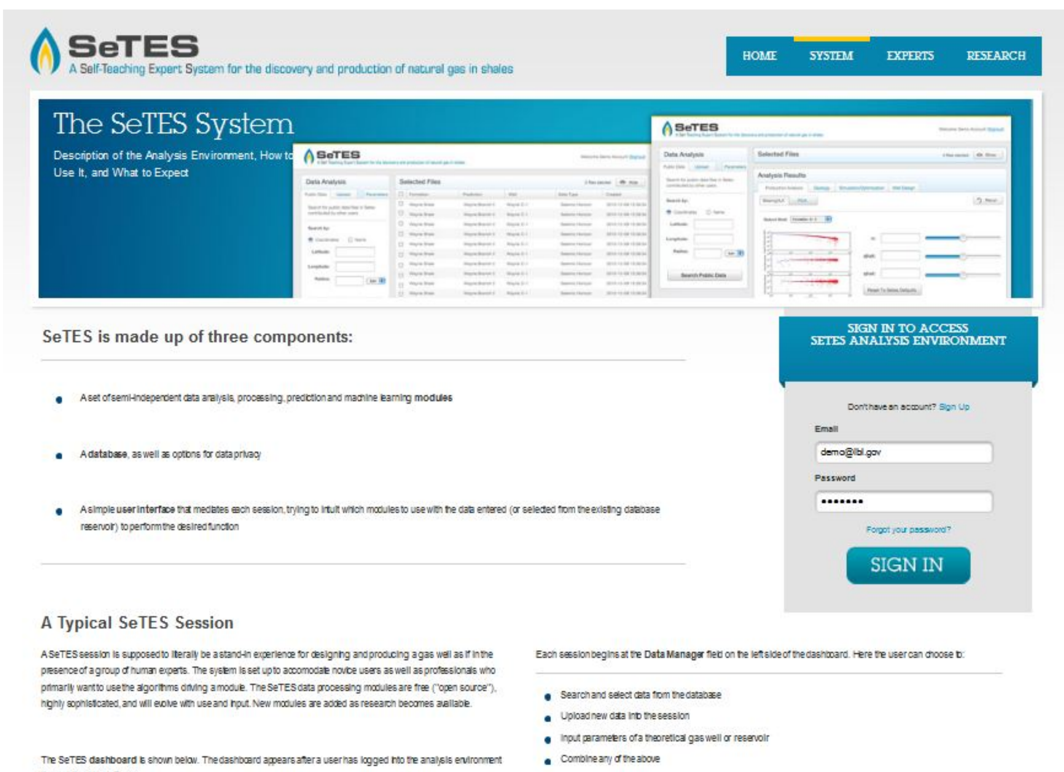

Figure 3: System Page.

deemed relevant and useful to users.

From the SeTES Homepage, the secondary tabs at the top of the page link to the **System Page (**Figure 3**)**, the **Experts Page (**Figure 4**)** and a **Research Page (**Figure 5**).** The Experts Page lists the names, titles, and affiliations of the various people that have written the modules and made them available on the SeTES website. The page title, "Experts," was chosen over Specialists, Professionals, or Our Team to better reference the "Expert System" of the SeTES acronym. This page also helps to show how the SeTES expert system represents a group of different experts from different fields who contribute more than one type of expertise. Currently the Experts Page does not link each person to the module he or she produced, but it should make clear the different groups of specialists that are associated with those modules (for example: geologists, petroleum engineers). The Research Page contains a list of all papers, presentations, and theses produced by the SeTES project, both in development and as an ongoing website entity. In addition to listing the titles of these papers and presentations, the Research Page includes links to the research as PDF files, so that they are readable immediately and in their entirety. The Contact Page includes a contact form as well as email addresses, to encourage visitors to the site to respond, ask questions, or submit feedback. We may add a discussion forum or open comment field in the future if interest seems to justify it.

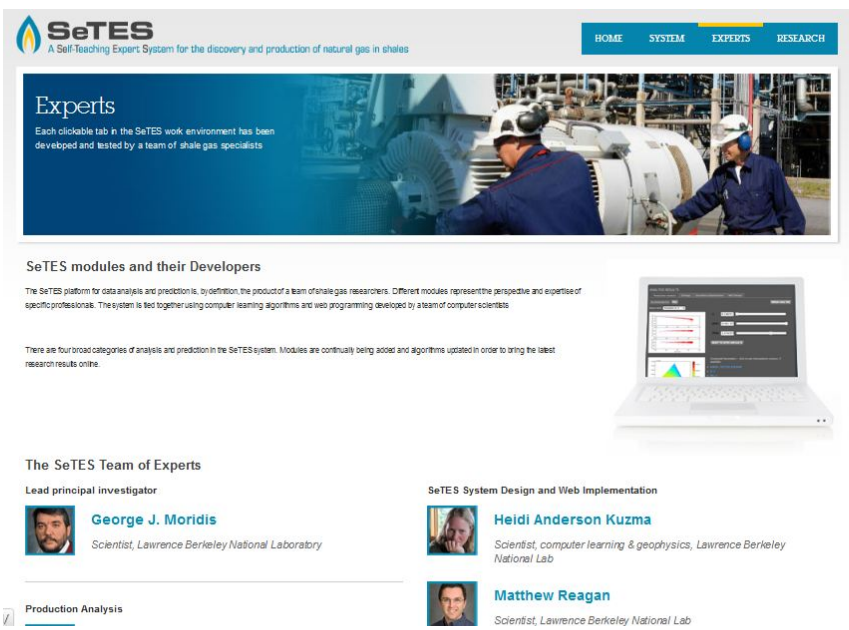

Figure 4: Expert Page.

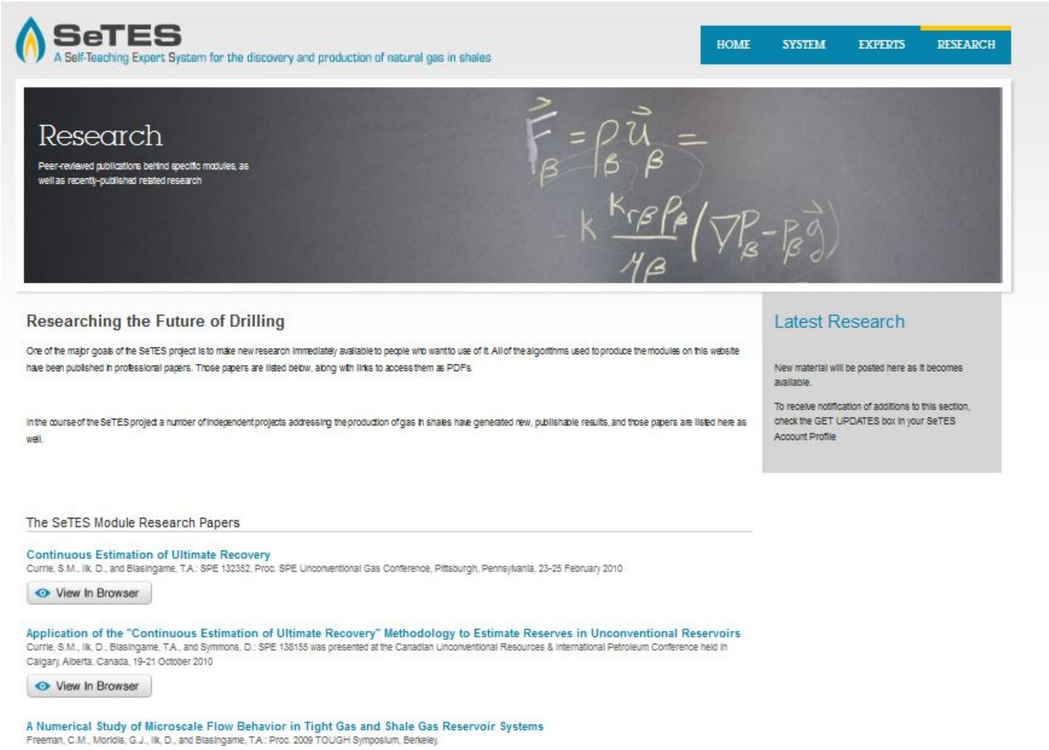

Figure 5: Research page.

4.1 A SeTES walkthrough

4.1.1 Signing In/User Accounts

The first step to using the Analysis Environment (which includes the database and all of the performance modules) is to sign in with a username and password at the **Sign In** box on the Homepage. Anyone can request an account. The login serves primarily to segregate users into unique workspaces, each with its own secure private database. Within an account, SeTES also preserves the selections used in previous visits, allowing easy replication of analyses. Other functions of user accounts include:

- To collect email addresses for future notifications of changes and updates to the site
- To keep data on how well-used the site is, and what data is most popular
- To create a minimum level of screening for seriousness of purpose

The sign-in box appears on the Homepage, as well as on the System Page. Signing in is not required to explore the site and access the information pages, but it is required to get to the Analysis Environment and to use the expert system tools.

Signing in takes the user directly to the analysis **Dashboard** Figure 6, which is the center of the SeTES analysis environment. It consists of the **Data Manager** (at left), **Selected Files menu** (right) and **Analysis Results** (bottom)

4.1.2 Data Manager and Data Privacy

In the data manager, to the left of the screen, the user sets up the data and initial parameters for the session. There are three ways to enter or select data, accesses by clicking on one of the three tabs. Any combination of the data-entry, selection or parameter setting modes may be used.

Database Tab (shown Figure 6): The user can search the existing public database based on a coordinate location, a specific well name or the name of a specific field (such as Holly Branch) or reservoir (such as Bossier Shale). SeTES will return all of the data that fits the query. At this point, the user can select all data, or a subset of what has been provided by selecting the appropriate boxes in the Selected Data menu box.

Upload Tab (Shown in Figure 7): The user can upload data in a variety of industry-standard data formats. The accepted data types are production logs (rate and pressure), geophysical well logs (*.las), well-bore diagrams (text files), geophysical horizons (text files), and shut-in test data (text files). At the time of upload, a user can elect that the data

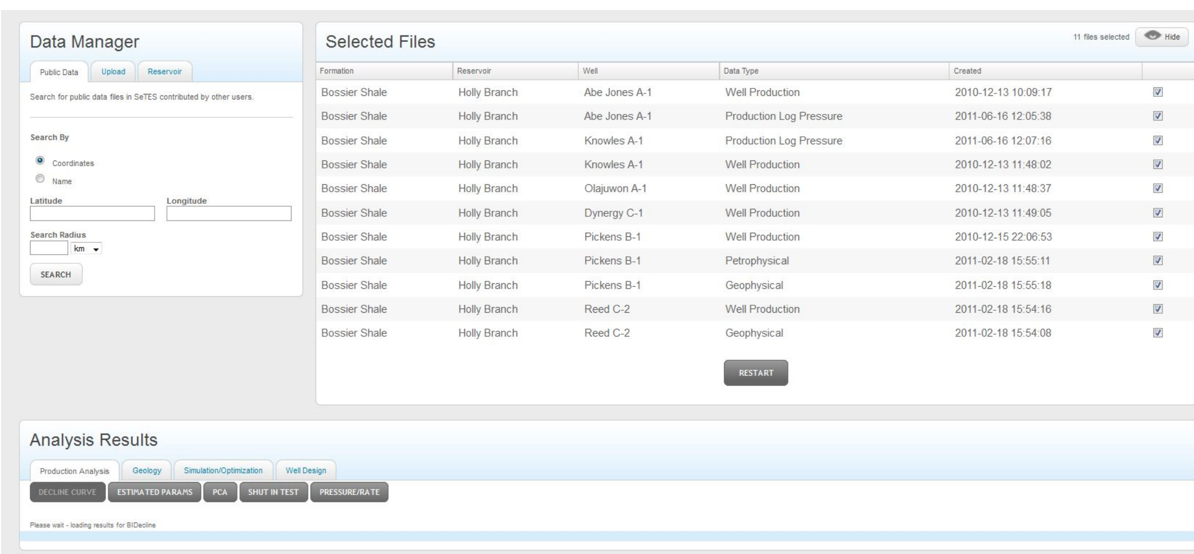

Figure 6: Dashboard

be used by SeTES in one of three modes

- *Public* : The data will be made available to the SeTES database at the end of the session
- *Semi-public* : All specifics of the uploaded data will be erased at the end of the session, but results obtained during the session may be saved to the SeTES database.
- *Private* : At the end of the session all of the data and parameters derived from it will be erased from the SeTES server.

If the user chooses either of the first two options, the data he enters will become a part of the SeTES database and be usable either for general analytics (prediction mode) or accessible by other users interested in that well formation or field. The choice of *semi-public* means that specifics of the information such as location and names of wells are deleted but statistics are kept on parameters derived from the data. If the user chooses to keep data *private*, all of the data uploaded will be deleted from the server at the end of the session; none of the information will be added to the system or made available publicly, even for analytics.

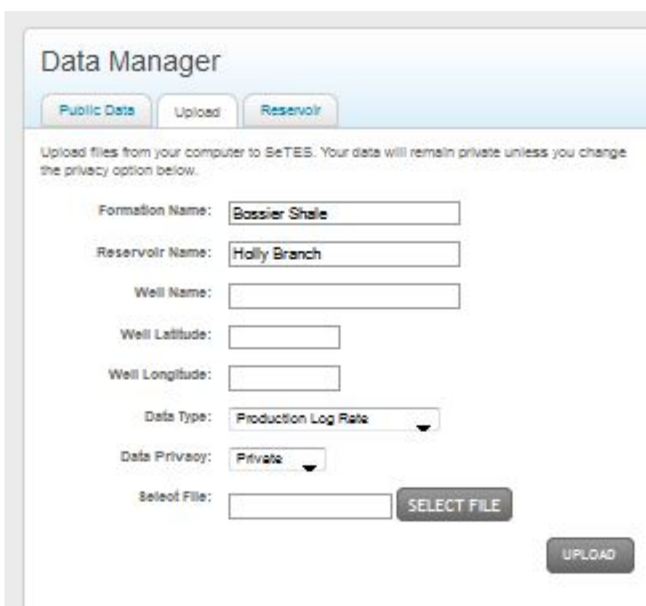

Figure 7: Data Manager Upload Tab.

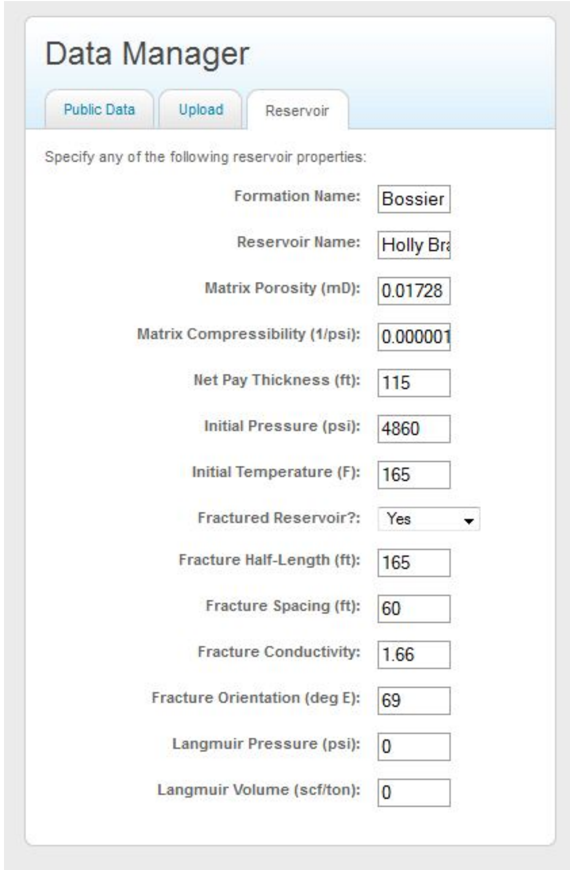

Figure 8: Data Manager Reservoir Tab.

Reservoir Tab (shown in Figure 8): The third option for the user is to upload no specific data, but to instead to input values for petrophysical parameters such as permeability, porosity, or net pay thickness. These values may come from experience or be purely speculative. SeTES will use these values in its analysis.

4.1.3 Analysis Modules

Once all of the desired data have been uploaded, the files selected, and any parameters entered, the analysis will begin as soon as the user clicks on the **start analysis** button. The system will automatically run the first module and display the **analysis results** as an interactive set of tabbed pages.

The tabs, shown previously at the bottom of Figure 6, are arranged in four module groups. When the user clicks on one of the module groups, sub-tabs for all of the modules in that group appear below it. Each of these modules is designed to perform a specific data analysis task that may, or may not link to tasks performed in other modules. A full description of all of the modules is given in section 4.1.3 Analysis Modules.

4.1.4 Exit

Currently the exit mechanisms for the system are under development. Implementation is pending usability testing. However, we envision that, having run the program modules and produced the requested results ("report"), the system will ask the user where he wishes to go next, offering the following options:

- Save report and exit (Sign Out)
- Download/export report
- Print report
- Save report and run a different report with the same data
- Save report and return to Data Manager to select or enter new data
- Exit (Sign Out) without saving report

The default system response if the user navigates away from the SeTES website at this point is to delete all reports. If the user leaves the Dashboard for another page on the SeTES website (the System page, for example, to get guidance), none of the information entered is lost, and the back key will return him to the analysis environment dashboard.

5. Experimental Methods

In this section, we discuss the inner workings of the system, starting an overview of the Holly Branch data set which we used to develop the system. The we give technical descriptions of all of the modules, including those currently under development. We then move to a technical description of how the SeTES software is organized, ending with a description of several novel ideas that were developed for SeTES but are not yet integrated into the Alpha-system.

5.1 Holly Branch Data Set

In order to build the system, SeTES partner Anadarko Corp. courteously provided an extensive initial data set from the Holly Branch field in Freestone County, Texas. An overview of the field area is provided in Figure 9.

The dataset includes:

- Daily Production Rate data from 32 wells
- Daily Production Pressure data from 8 wells
- Geophysical Well logs from the same 32 wells
- Picked seismic depths for 8 horizons (none within several hundred feet of the pay zones)

- Geologic cross sections for some of the pay zones
- Bit records from the 32 wells
- Mud records from the 32 wells
- Misc. treatment records for a subset of the wells
- Well completion reports for the 32 wells
- Well bore diagrams for the 32 wells
- Core studies from 7 wells

From this data, we identified a set of 12 "core" wells which had clean, well-behaved production data to serve as the initial data for analysis. The decline curve behavior of the core wells is plotted in Figure 11..

Altogether, the Holly Branch dataset comprises about 1.27 GB of data. It is a small amount of data compared to what eventually will be stored in SeTES, but it is an excellent starting point because of its completeness and variety.

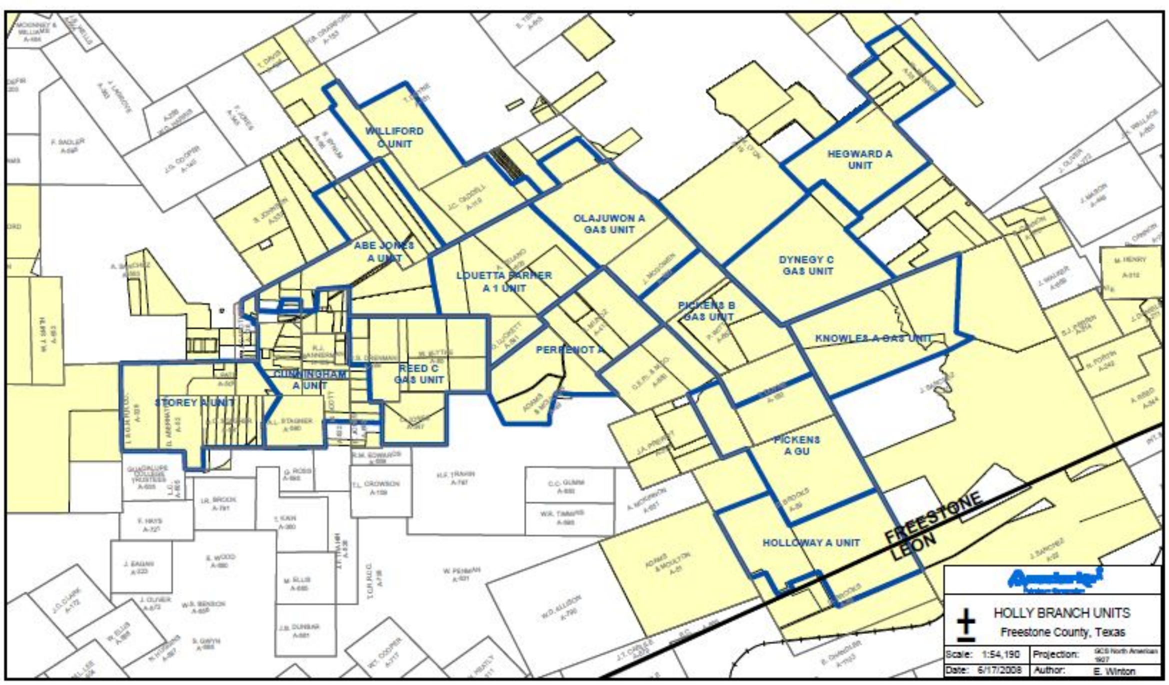

Figure 9: Holly Branch location map

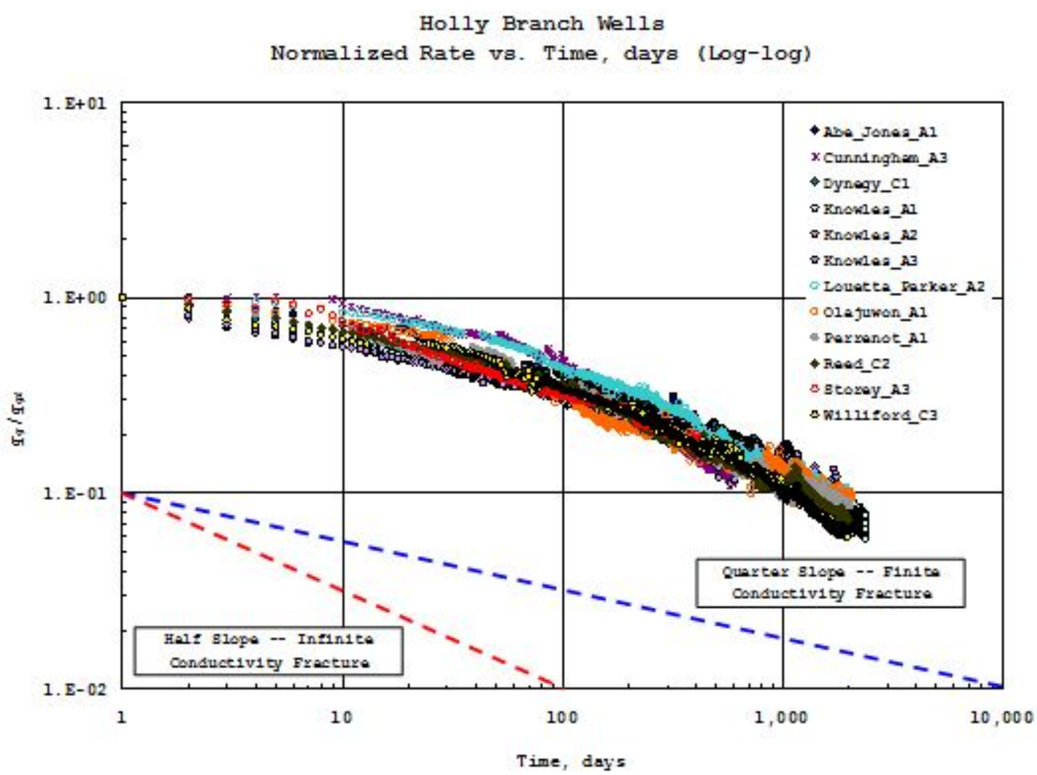

Figure 11: Decline curves for Holly Branch Core wells.

5.2 Modules

This section includes a description of each SeTES module. As the modules perform different functions, require different datasets (at times), and were developed by different teams of researchers, they represent a variety of experimental methods aimed at producing complementary results. The modules are grouped into four categories on the Analysis Environment page of the website. They are described in this section by order of appearance on the web page, from left to right.

5.2.1 Production Analysis Module Group

The modules in the Production Analysis Group compute analyses that are directly related to data that is measured during production, particularly rate and pressure data (decline curves) and data from pressure tests. It is through these modules that estimates of petrophysical parameters directly related to production are established.

5.2.1.1 Decline Curve Analysis

Required inputs:

- *Production rate files from one or more wells.*

Outputs:

- *Decline curve parameters as estimated using Blasingame/Ilk analysis*

SeTES uses the empirical "power-law exponential" rate decline relation [*Ilk, et al.* 2008] for decline curve analysis. The decline relation is given as:

$$q(t) = \hat{q}\exp[-D_\infty t - \hat{D}t^n] \qquad 1.$$

$q(t)$ is production rate measured in MCF/day, t is time in days and D_∞ is set to a constant value of 10^{-8}. The decline curve parameters, $\hat{q}$, $\hat{D}$ and n are determined for each decline curve selected by the user using a non-linear, least-squares fitting algorithm. The starting values for the fit are taken from average values of decline curve parameters for the selected formation or reservoir that are stored in the SeTES database.

Results from the parameter fit are shown in the Decline Curve module window (Figure 12).

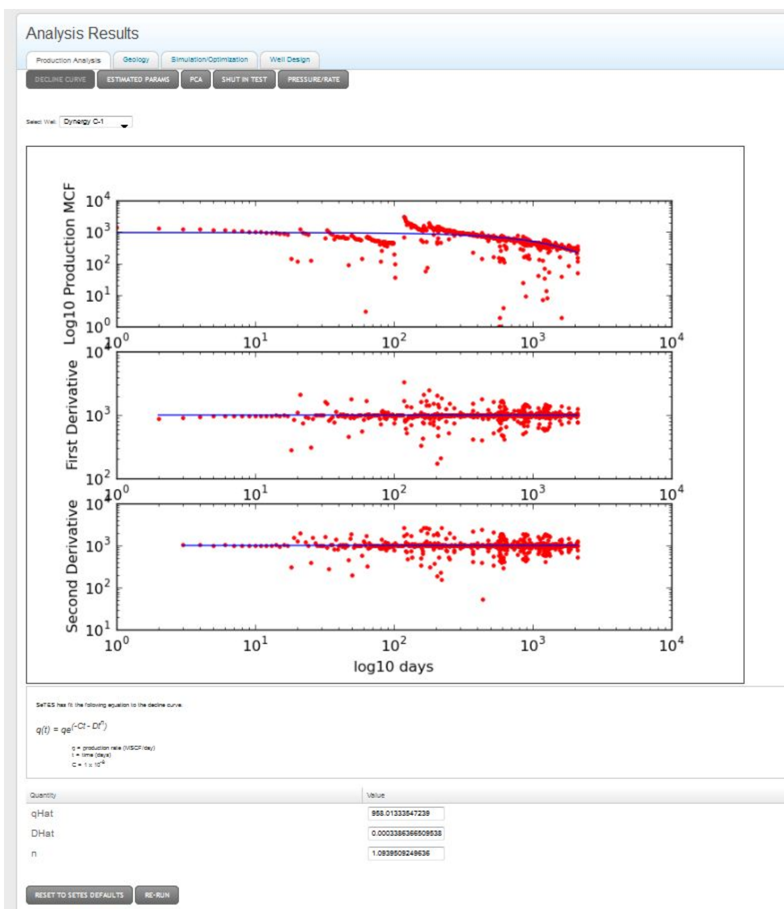

Figure 12: Decline Curve Module

If SeTES is unable to fit one or more decline curves, then the user will be notified and prompted to upload a cleaner version of the data file or deselect the file from the Data Manager. If no action is taken, the curve will not be included in further SeTES analysis.

Results for $\hat{q}$, $\hat{D}$ and n are presented in an interactive form. If desired, the user can change the values and click on the re-run button. If this is done, then the figure will display a decline curve based on the entered values (Figure 13). Parameter values used in subsequent modules will be the ones that the user has set. At any time during the session, the user can click on the Reset to Defaults button to restore the SeTES computed values.

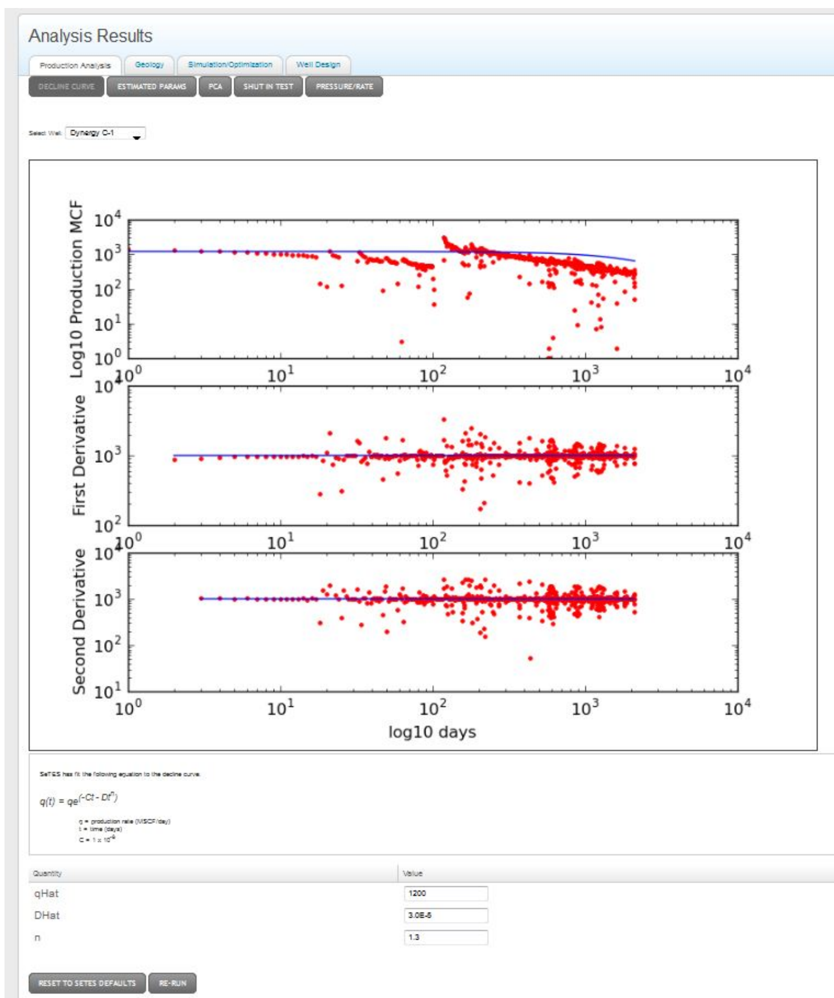

Figure 13: Decline Curve re-run mode.

Note that once the user has selected new values for the decline curve parameters, the curve is not fit as well as with the automatic fit.

In the original design of SeTES, all of the curve fitting was going to be done using a Jackknife algorithm to estimate uncertainty [Brillinger, 1964]. The Jackknife works by fitting the entire dataset once and then fitting a number of subsets of it, throwing out data in turn. If part of the dataset contains outliers, the final results of a jackknife will tend to be less biased. Furthermore, an estimate of the standard deviation of the estimates made using partial datasets. The downside of the method is that it requires many iterations of curve fitting and is computationally intensive. Although we were able to run it in test versions of the system with one or two decline curves, its use leads to lengthy delays and potential browser timeouts. It remains fully implemented in the system, but unused.

5.2.1.2 Estimated Parameters

Inputs from other modules:

- *Decline curve analysis: Decline Curve parameters. If possible, Shut-In test or Rate-Pressure analysis: Estimates of fracture half-length and permeability otherwise values from database are used.*

Outputs:

- *Estimated Ultimate Recovery, Permeability and fracture half-length for each well*

SeTES uses the results from the Decline Curve analysis to estimate Ultimate Recovery, permeability, and approximate fracture half-length for each of the wells for which decline curve parameters were estimated. If no decline curve data has been selected, SeTES will prompt the user to select appropriate data in order to run the module. Estimated Ultimate Recovery (EUR) is computed by projecting the rate of production forward 15 years using the decline curve relationship in Equation 1 and parameters estimated by the Decline Curve module. The cumulative production is sum of predicted daily production. This method does not take into account downtime in production.

Permeability and fracture half-length are estimated from the decline curve parameters using the following empirical relationships developed by Blasingame and Ilk (Details of the method are given in the report in Appendix II).

$$k = an^b \hat{D}^c \hat{q}^d \qquad 2.$$

k is permeability, $\hat{q}$, $\hat{D}$ and *n* are the decline curve parameters and *a,b,c,d* are parameters which we describe below.

Fracture half-length, x_f, is estimated using

$$k\,x_f = \alpha \exp[\beta\, n] \qquad 3.$$

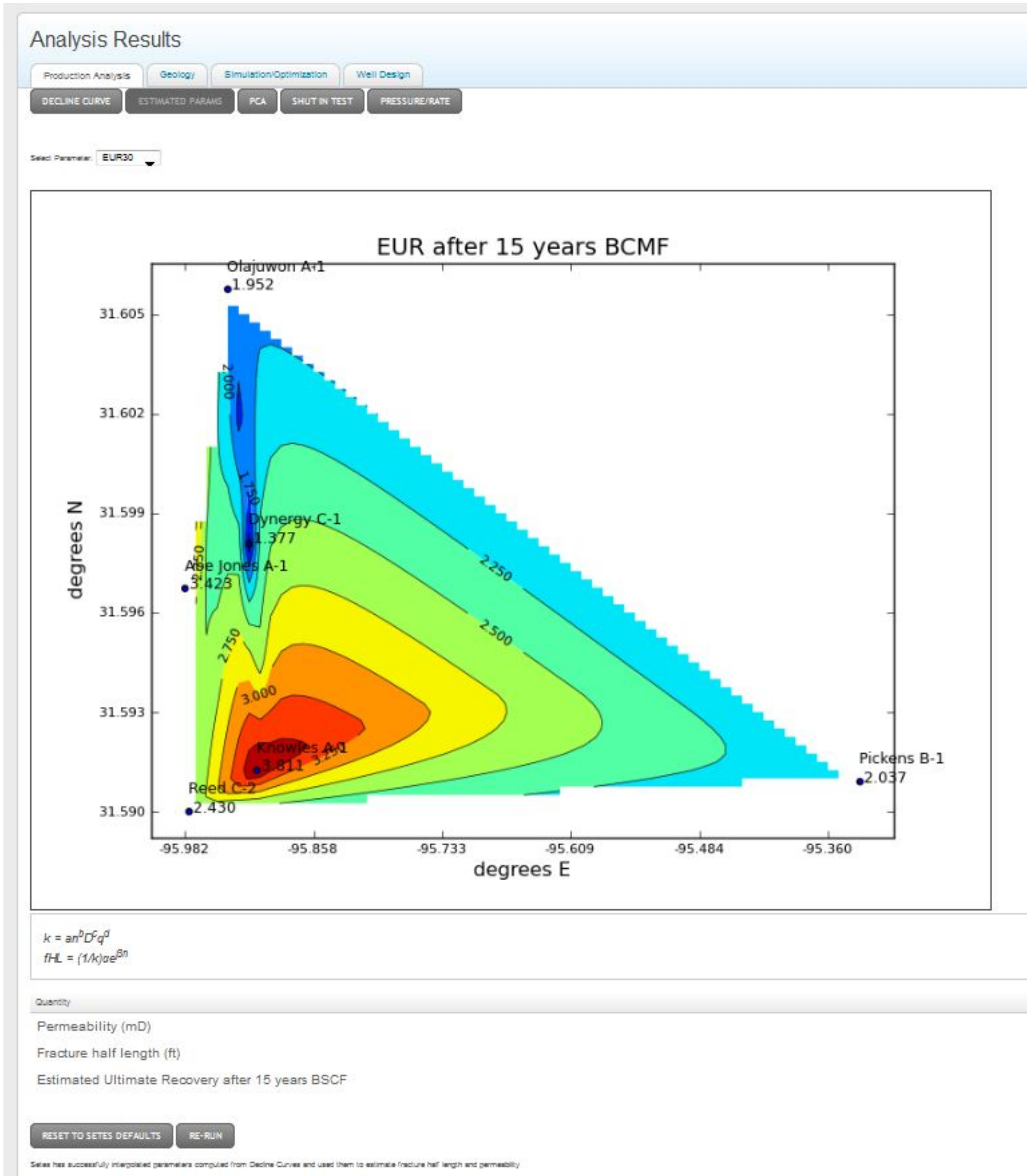

Figure 14: EUR interpolated between wells.

In order to use these relationships, it is necessary to have estimates for the conversion constants *a*, *b*, *c*, *d*, α, and β. Initial values were derived from fitting the equations to data from the Holly Branch field, where estimates of *k* and x_f had been made using a model-based technique that takes into account both decline rate and pressure (again, details are given in Appendix II). A module which implements such a model-based analysis is under development for SeTES, as well as a module which will estimate the conversion constants if rate/pressure data or data from a shut-in test is available (see 5.2.2.1 Rate-Pressure Analysis).

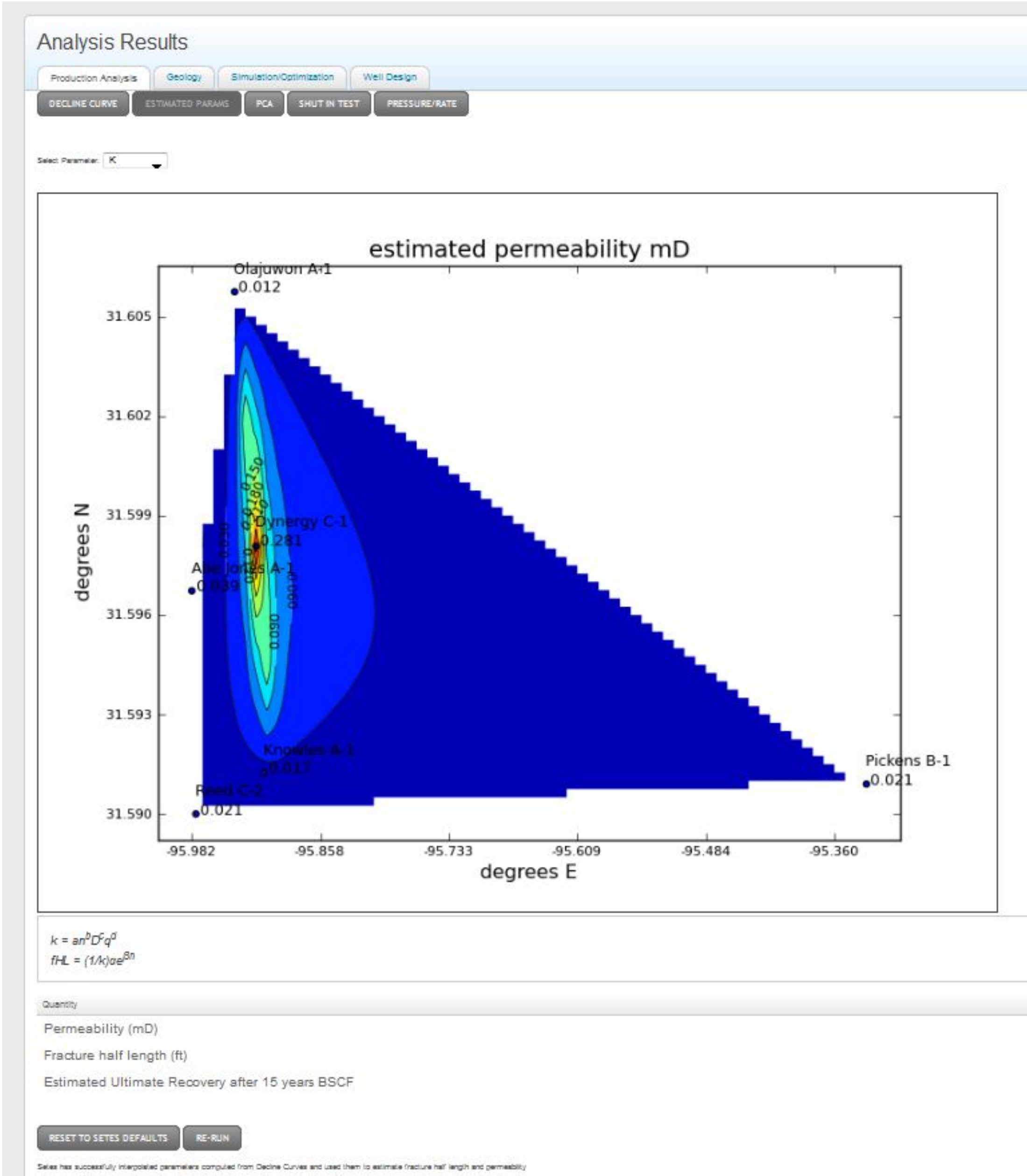

Figure 15: Estimated Parameters, Permeability.

Once an estimation of the conversion constants has been made, the results are stored in the SeTES database for use in other analyses in which pressure data has not been made available. As SeTES sees more data, its estimates for these constants should improve. This is one of the key self-teaching features of the system.

After estimates of the three parameters have been made for each of the wells, SeTES interpolates between them, choosing an algorithm based on the number of wells that have been selected. If that number is four or less, linear interpolation is used. If the number is five or more, cubic spline interpolation is used. (This is standard for all of the interpolations done in SeTES). Results are illustrated Figure 14, Figure 15 and Figure 16.

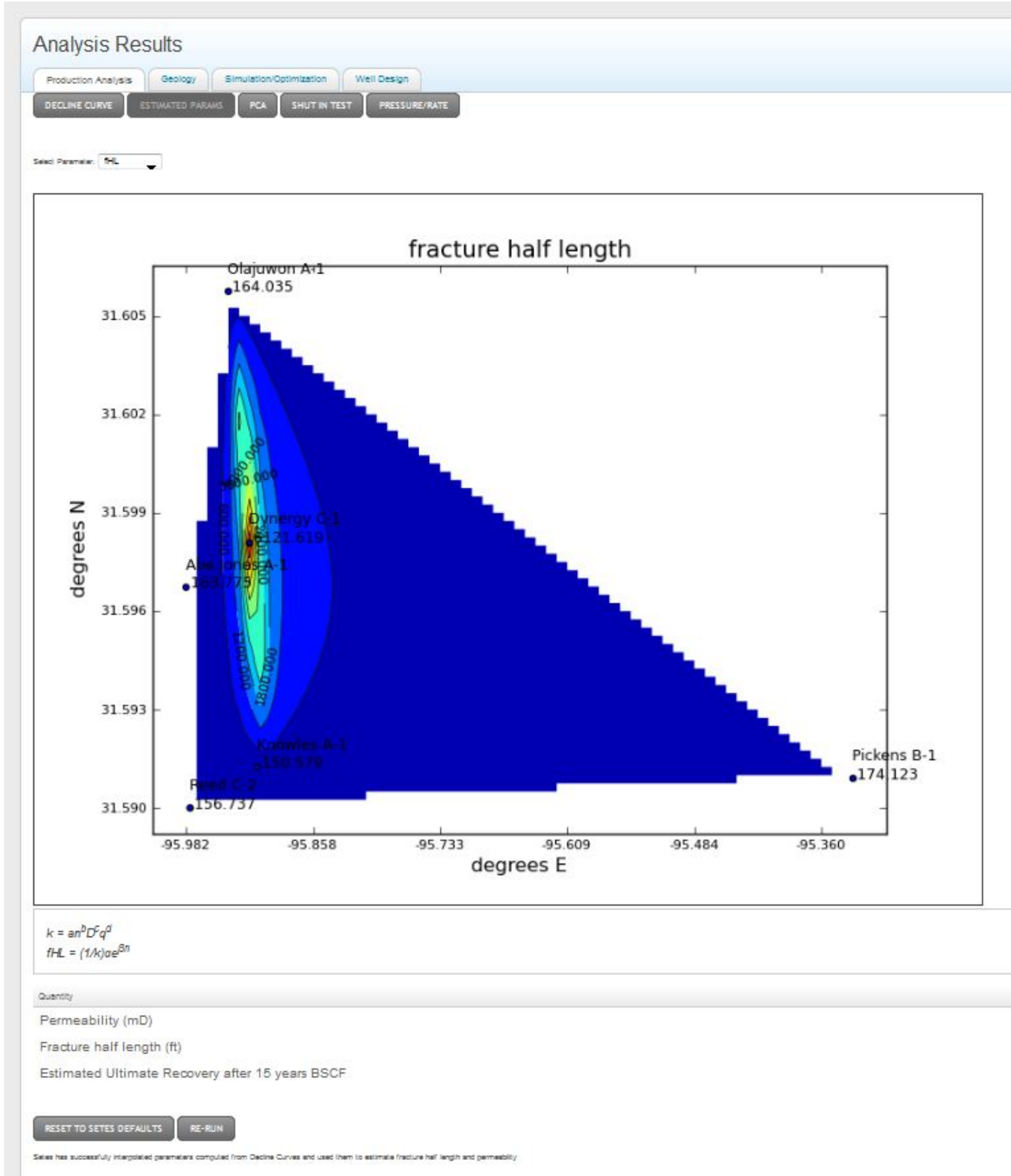

Figure 16: Estimated Parameters, Fracture Half Length.

5.2.1.3 Principal Component Analysis (PCA)

Required inputs from other modules:

- *Decline Curve Analysis: Decline Curve Parameters*

Outputs:

- *Principal component scores for each well included in the analysis.*

Principal Component Analysis (PCA) is used to sort wells that have gone through decline curve analysis into groupings based on production behavior. The result is a plot such as

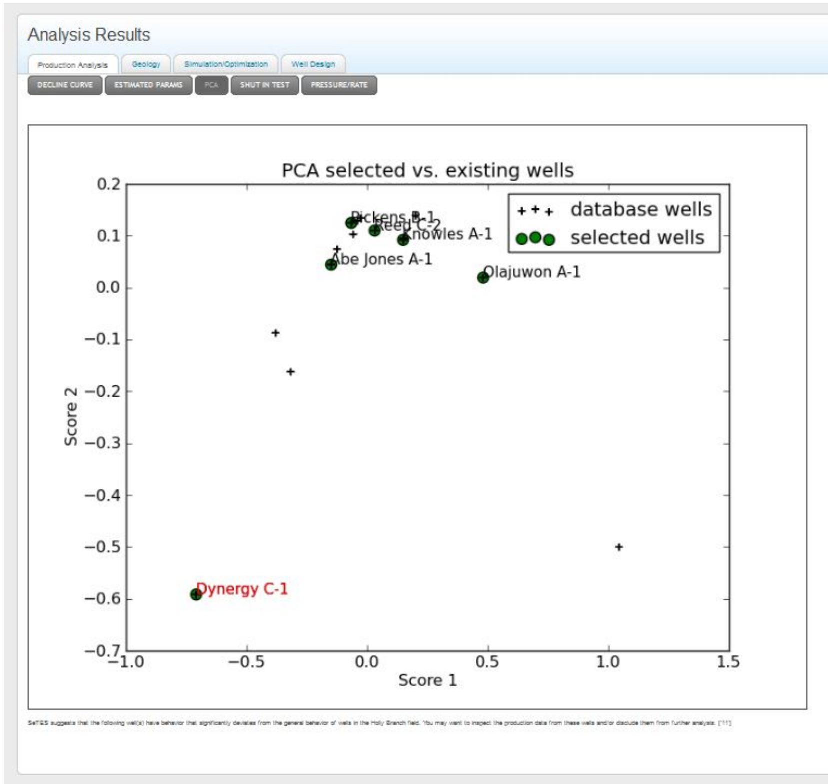

Figure 17: Principal Component Analysis Module

Figure 17, in which the selected wells are plotted on a 2D scatter plot to indicate similarities and differences. Wells with similar decline curves will land in similar locations on the figure. If decline curve parameters from nearby wells are available in the public database, they are plotted as well for reference.

PCA is a powerful technique used commonly in many types of data analysis other than clustering. There is an excellent introduction to the subject on Wikipedia (http://en.wikipedia.org/wiki/Principal_component_analysis) as well as a number of other tutorials online. In particular the student tutorial by Smith [2002] gives an intuitive overview for general readers with some technical background.

If a set of observations (such as decline curves) are correlated or otherwise similar, then PCA will seek to explain each of them in terms of a set of patterns that the observations have in common. It selects these patterns such that the patterns themselves are uncorrelated with each other. These patterns are the *principal components.* The projections of data onto the principal components are known *scores*. Often the projection of data, such as decline curves, onto principal components plots can be thought of intuitively as measures of the amount of two or three patterns that describe each observation. For example, Figure 17 shows a plot of a number of wells based on a PCA of their decline curves and related parameters. Most of the wells have a lot of Pattern 1 and a little bit of Pattern 2 making them plot at the top of the graph. The decline curve of well Dynergy C-1 does not contain much of either pattern so it plots away from the rest in the lower left of the figure.

By examining the decline curve data from the decline curve tab, it is possible to determine by eye what is different about the well. The production curve of Dynergy C-1 is shown along with that of Abe Jones A-1 (a more typical well) in Figure 18. In this case, Dynergy C-1 appears to have a lower initial production rate than the other wells and a slowly tapering production tail. If a well falls too far away from the others (more than two standard deviations from the mean value of the group), then SeTES will suggest that either its data be examined or it be discarded from further analysis. It has to be removed manually by the user, however—by default, all data is retained. Otherwise, the results of PCA are not used

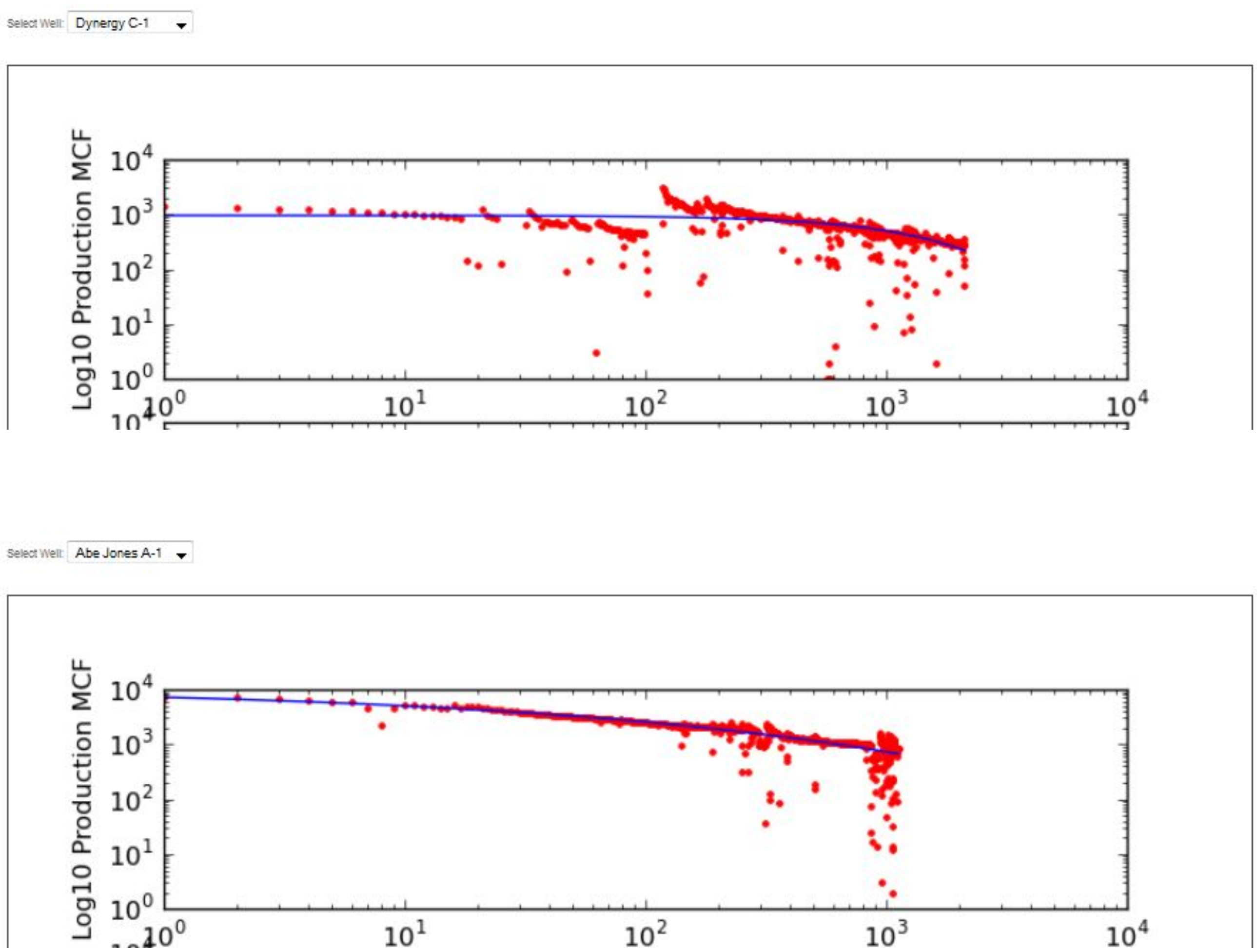

Figure 18: Production Curves for Dynergy C-1 and Abe-Jones A-1.

This illustration shows the difference between Dynergy C-1 and a more typical well, Abe-Jones A-1. The Dynergy curve is flatter than the Abe-Jones Curve and has a lower initial production. This may explain why it plots below the typical curves on the PCA plot.

directly in any of the other SeTES modules and may not be of seem to be of much interest to most users. However, future modules have been envisioned which use the technique. These include a module for projecting well-log parameters along 2-dimensional seismic sections (5.2.5.2 2-D Interwell Projection) and a module which has been requested by several industry representatives which would perform the otherwise laborious of automatically sorting through well-bore diagrams and production records to find wells with a similar treatment history.

5.2.1.4 Shut-In test analysis

Required inputs:

- *Build up or draw down data containing time (hours) and pressure or pressure drop (PSI) from at least one shut-in bottom-hole pressure test.*

Outputs:

- *Estimates of permeability and fracture half-length*

The Shut-In module uses the Fortran 90 program PERANA developed by Blasingame [1996] to calculate values for permeability and fracture half-length that best fit shut-in test data using a least-squares fitting algorithm. The software is designed to provide a "history match" of a given set of pressure-transient test data using analytical solutions for un-fractured and fractured wells in infinite-acting and bounded reservoirs via a Levenberg-Marquardt least squares algorithm. All solutions are given in the Laplace domain for convenience, and the Gaver-Stehfest algorithm is used to perform the required numerical inversion. The software can match on any combination of the following functions: pressure drop, pressure drop derivative, pressure drop integral, or pressure drop integral-derivative. The software utilizes solutions for both homogeneous and naturally fractured reservoirs. PERANA has the capacity to analyze data from both a drawdown and a buildup test, although it is currently only implemented in buildup mode in SeTES. As a stand-alone code, PERANA will estimate permeability, fracture half-length, skin-factor, a dimensionless wellbore storage coefficient, fracture conductivity, and an interporosity flow coefficient, fracture storativity and drainage radius. Currently only permeability and fracture half-length are implemented in SeTES.

The results from Shut-In analysis are displayed as fitted curves and interactive form(Figure 19). If the user decides to update or explore parameters, he can change the values and click `re-run` to return the system prediction mode and generate a new fit to the data based on the new parameters. The original parameters can be returned by clicking on the `Reset to Defaults` button. If Shut-In data is not available for analysis, then SeTES automatically switches to prediction mode and runs PERANA to estimate what the data from a Shut-In test would look like. Because no Shut-In data was included in the Holly Branch dataset, the SeTES-alpha implementation operates only in prediction mode, although the infrastructure exists for optimization mode. It has only been tested on synthetic data in this mode.

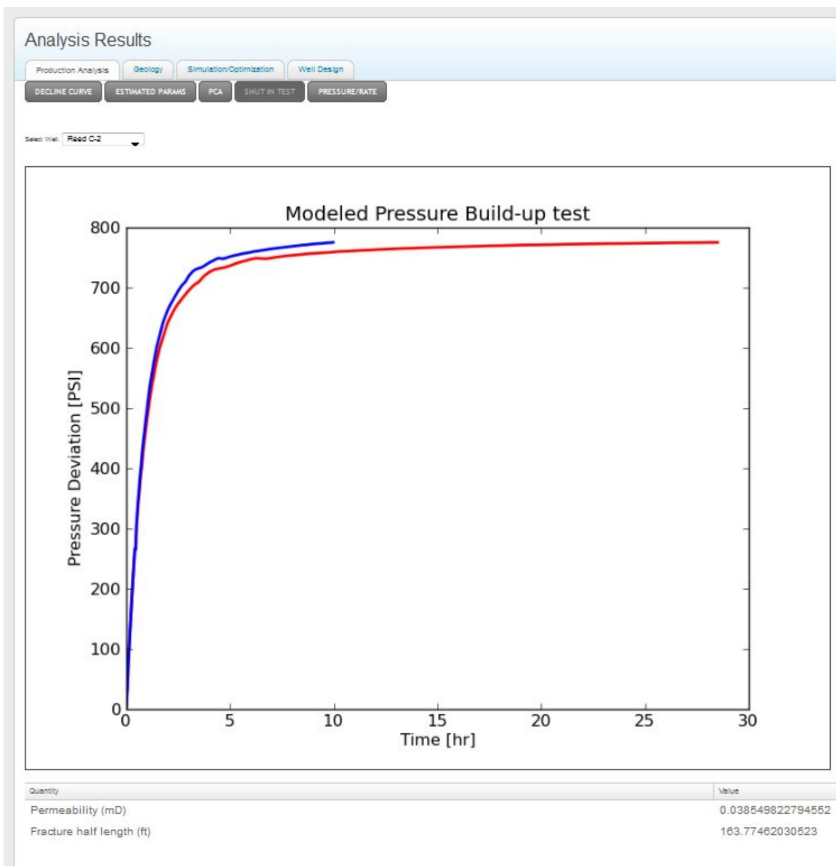

Figure 19: Shut-In test module

Results from a synthetic shut-in test (blue) are fit using the Fortran modeling program PERANA. The fitted results are shown in red along with an interactive menu showing results for fitted parameters permeability and fracture half length. As in other modules, the values can be changed by the user and the module can be re-run to reflect the new values.

5.2.2 Modules under development for production analysis

SeTES is still a system in Alpha development, which means that it is fully designed and partially implemented. (See section 6.1 State of the SeTES system). Along with all of the modules which are mostly or fully deployed in the system, we list modules which have undergone a serious development effort and are in the late stages of being prepared to be integrated with the system. (In most cases, the programming for the modules is completed and tested, and all that remains is plugging them – which is a complicated procedure, see section 5.4 SeTES technical implementation).

5.2.2.1 Rate-Pressure Analysis

Required Inputs:

- *Production curves showing rate and pressure for at least one well*

Outputs:

- *Estimates of permeability, fracture half-length and other parameters.*

Currently we are designing a module capable of estimating petrophysical parameters using a rate-pressure analysis. We have not decided on which algorithm we will use and if the implementation will be in Fortran or directly in Python.

5.2.2.2 Automatic Transient Detection

After production is shut down for a period of time, wells tend to re-pressurize and production resumes above the expected decline curve and then decays exponentially back to the normal level. The shape of these transients may contain important information about petrophysical parameters and in any case tends to corrupt the overall decline curve analysis. In the process of working on an (as yet) uncompleted module for automatic data clean-up, we implemented an algorithm which is capable of successfully locating these transient curves and estimating an initial production and time constant from them. The algorithm is too slow for inclusion in the current version of SeTES; however, we hope to be able to include it when the system is deployed on more powerful, cloud-based servers.

The algorithm works by subtracting the measured data from the best fit of Equation 1 to the decline curve. It then locates peaks by looking at the numerical derivative of the residual. At each peak location, it attempts to fit an exponential curve of the form:

$$q(t) = A\exp[-bt] \qquad 4.$$

Where A is the amplitude of the peak above the normal decline curve and b is a time constant. In order to find the end of a transient, it starts at the beginning of the next transient and then fits Equation to successively shorter periods of data. It chooses as the transient the data series that can be fit with minimum error. An example is shown in Figure 20. Once the module has found the transients, it plots them along with the decline curve, along with the amplitudes and time constants. Then, it fits a straight line through the time constants to determine if they are increasing or decreasing with time. Since these time constants have a relationship to permeability, this technique might be useful in attempts to study long-term permeability changes as a function of production time.

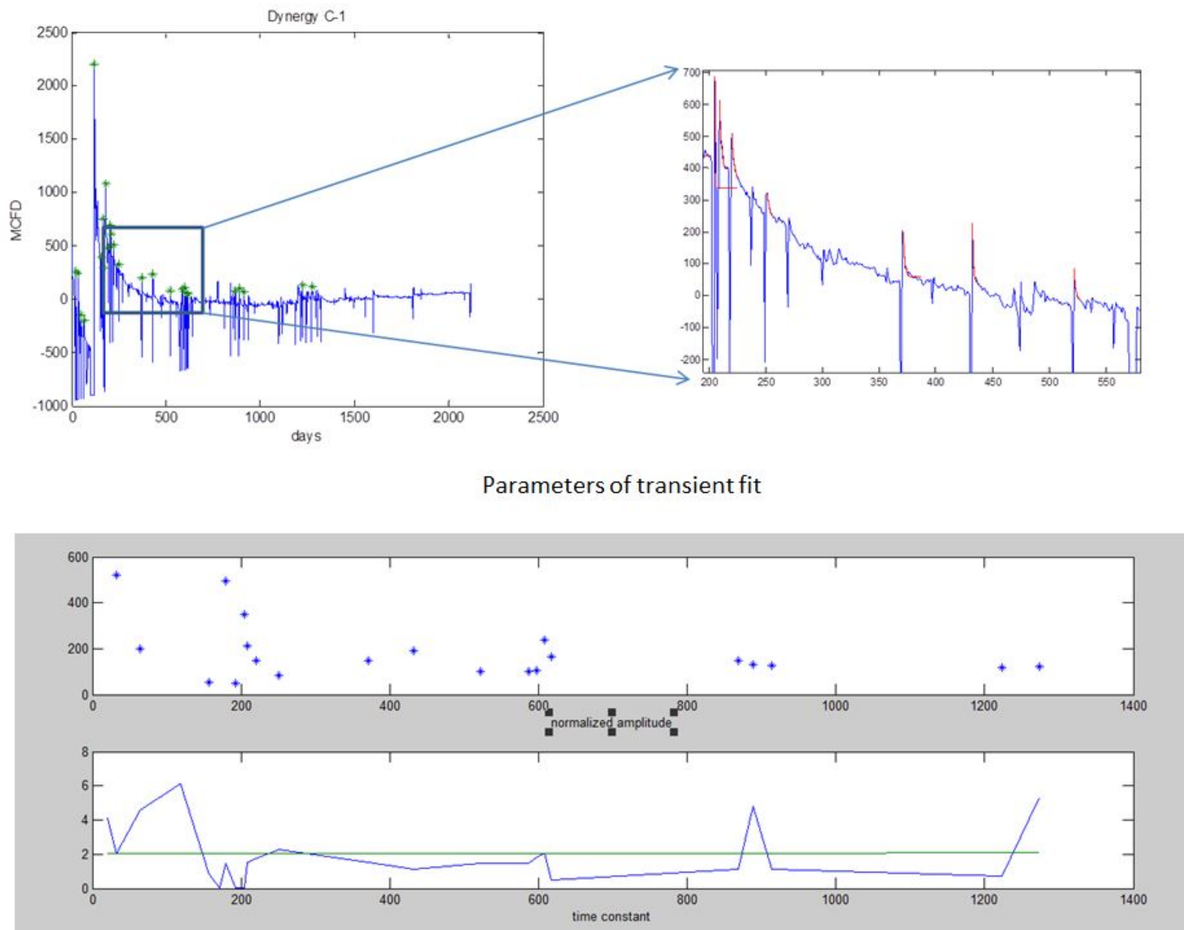

Figure 20: Automatic Transient Detection.

An algorithm designed to automatically detect the transient curves caused by re-pressurization after a well shutdown locates the peaks of the transients (green stars on top left figure) and fits an exponential decay curve to the transient (red curves on blow-up figure top right). The Amplitude and time constants of the transients are plotted as a function of time. They may be related to important reservoir properties such as time varying permeability.

5.2.3 Geology/Geophysics Module Group

The Geology/Geophysics module group contains all of the modules which are used to estimate petrophysical parameters of a reservoir in ways that are not related to Decline Curve analysis.

5.2.3.1 Unit

Required Inputs:

- *Well-bore diagrams from at least one well giving depth net pay thickness of named layers and stages*

Outputs:

- *Net pay thickness of all known layers that intersect selected wells*
- *Grids containing interpolated values for net pay thicknesses between the wells*

The Unit module interpolates net pay thickness between wells for which such data is available from completion reports and plots surface plots for all of the named layers and stages. For each named layer, it plots 2D figures of the interpolated layers. It also plots a 3D summary figure showing all of the layers. (Figure 21). At the moment, Unit is the simplest of all of the SeTES modules, however it produces the critical output for the Optimization and Simulation modules—namely an estimate of net-pay thickness of all of the layers that intersect selected wells.

Although the data manager contains a field for uploading completion reports, we have not yet written a utility to parse them and upload the information into the database directly. (For testing purposes, we entered the values into the database manually) We have considerable experience using templates in Python to read and parse input files, so this will not be a difficult utility to build.

The interpolation subroutine used to do the interpolation is the same as that used in the Interpolate Parameters module.

5.2.3.2 Well-log analysis

Required inputs:

- **.las files from one or more geophysical well logs*

Required inputs from other modules:

- *Unit: Top and bottoms of pay layers*

Outputs:

- *Estimated Porosity values for each of the pay layers intersected by the well.*

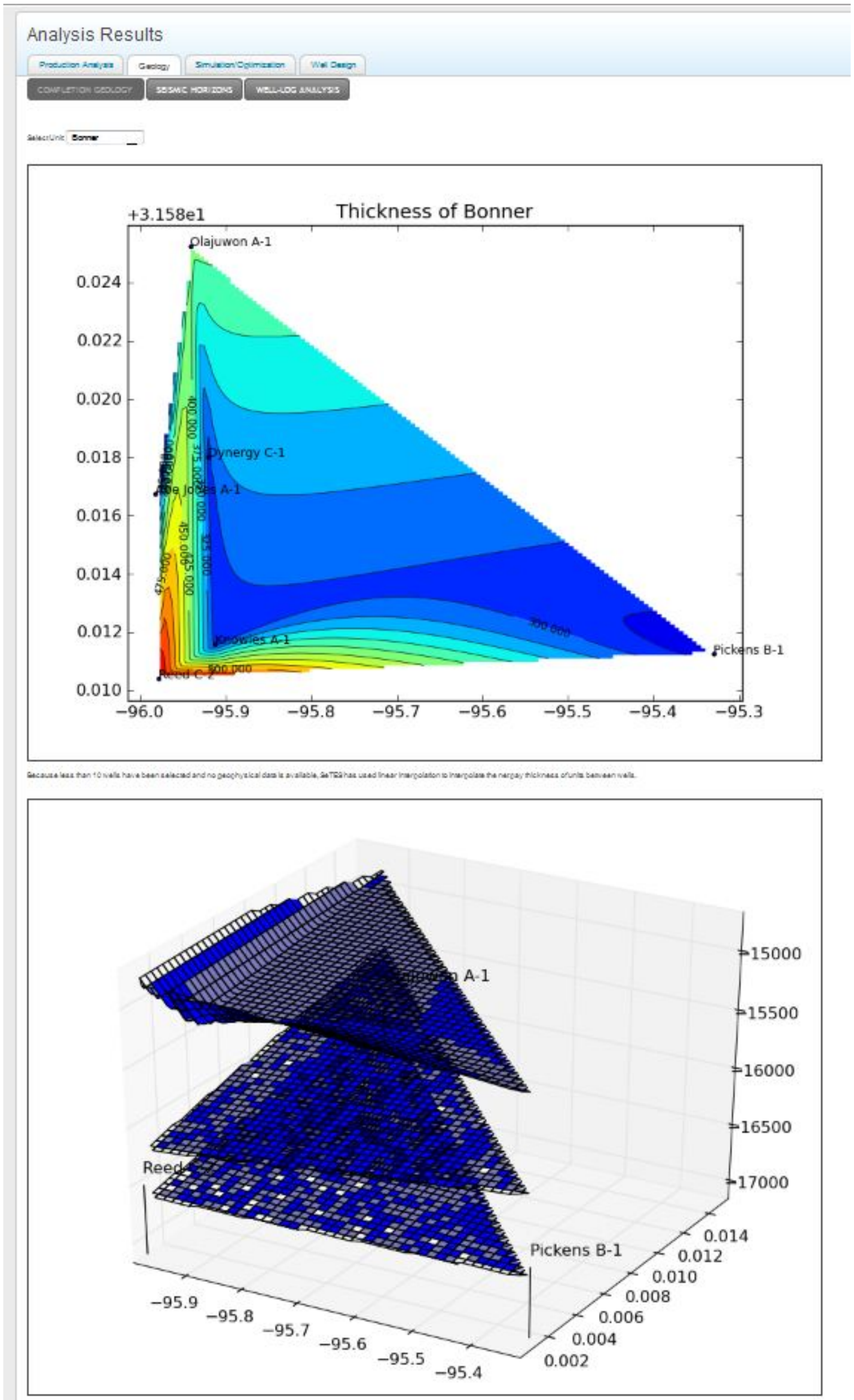

Figure 21: Unit module

The well-log analysis module automatically reads text files containing geophysical well-log data in *.las format. Currently it only reads the neutron porosity values from the files, however with little modification it could read any of the fields.

It uses the output from the Unit tab to find the top and bottom of pay layers in the file and then computes the mean and standard deviations of the porosity values. No attempt is made to find errors in the depths between the two sets of files, although this may be a feature to be added in the future. An example of the output from this module is shown in Figure 21.

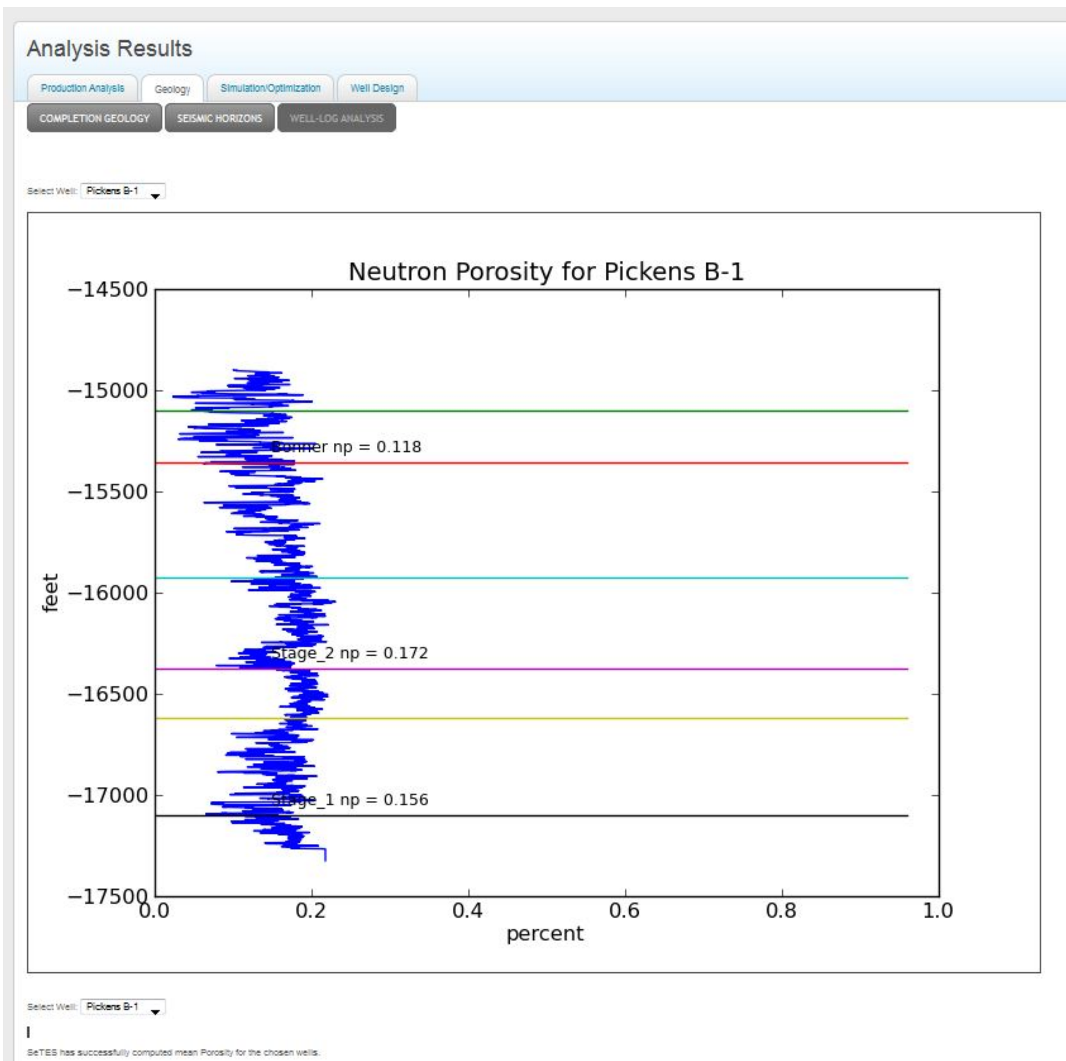

Figure 22: Well Log analysis module.

5.2.4 Modules Envisioned and under Development for Geology/Geophysics

The Holly Branch field has almost no seismic impedance contrast between pay zones and surrounding layers. Even though the next two modules were among the earliest to be developed for SeTES, they are not integrated into the system because we do not yet have the data to test them.

5.2.5.1 Seismic Horizons

Inputs:

- *At least one file containing xyz coordinates for a pre-picked seismic horizon*

Inputs from other modules:

- *Well Log: interpolated grids with values for porosity*
- *Unit: interpolated grids with values for thickness*
- *Interpolate: interpolated grids with values for permeability*

Outputs:

- *Grids containing xyz coordinates and interpolated values for porosity, permeability or thickness which have been projected along the seismic horizons, giving an estimate of depth to the unit and values for the parameters*

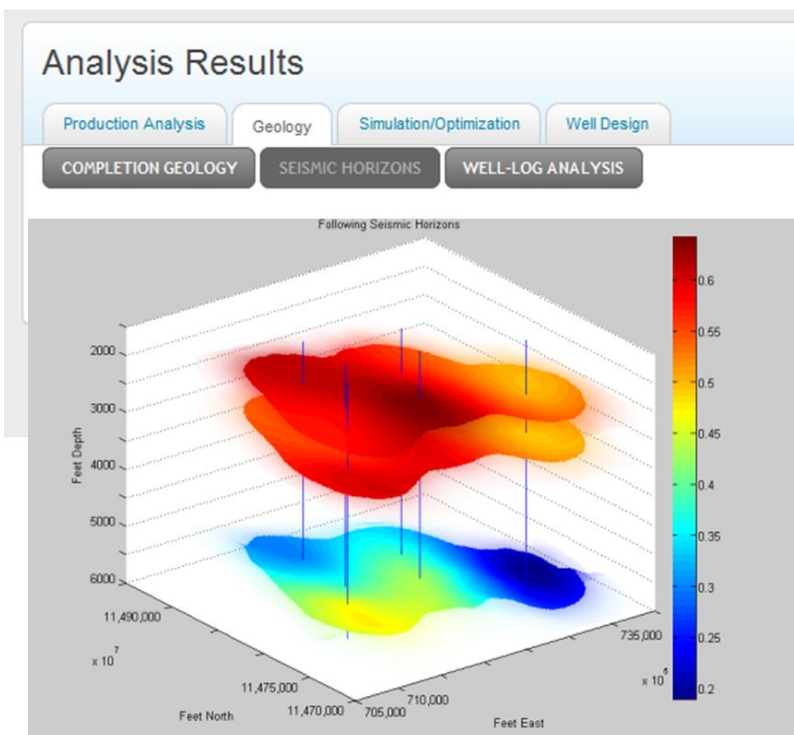

Figure 23: Seismic Horizons.

Given a text file containing location and depth of seismic picks along a seismic horizon corresponding to a pay layer, the module projects parameters interpolated between wells in other modules along those horizons. If the depth of the horizon does not match the depth of a layer given in the Unit output derived from well-bore diagrams, the module will assume that the velocity model used to make the depth/time conversion is faulty and adjust the seismic depths to match the well-bore depths, interpolating values accordingly.

This seismic horizon module was developed for an early version of SeTES; however, it got put on a back burner because there is no appreciable seismic impedance contrast in the pay layers of the Holly Branch data set. An example of the plots it generates is shown in Figure 23. .

5.2.5.2 2-D Interwell Projection

Inputs:

- *Geophysical well logs (*.las files) from at least two wells*
- *A 2-d seismic section or profile of seismic attributes between them*

Outputs:

- A *2-d image of plausible well log parameters interpolated between the wells using the seismic attributes to determine structures with plausible values*

This module was developed to interpolate well-log parameters between wells and generate a plausible 2-D image. The low-frequency content of the image (i.e. major geologic structures) is taken from structures in correlated 2-D seismic images as explained below. Higher frequencies (i.e. small scale variations in parameters such as porosity and electrical conductivity) are simulated by using a probability distribution learned from the well-log data. In this way, well log data and seismic data are fused. Details of the algorithm are given in Fernandez-Martinez [2010].

If seismic data or attributes are available to support it, the Projection algorithm works in the following way: Well log data is matched to a seismic 2-D profile.. Then the image of the seismic section is decomposed using an algorithm based on Singular Value Decomposition and which is directly related to algorithms commonly used in automatic image processing and classification. This means that the seismic section is reduced to a set of singular values and corresponding Eigen-images. In the final step, a new set of singular values is found by fitting the first few Eigen-images to the well-log data at the well locations instead of the seismic image. Since these images contain most of the structural information between the wells, the result is a projection of well log measurements along seismic structures. An example using synthetic data is shown in the top figures of Figure 24.

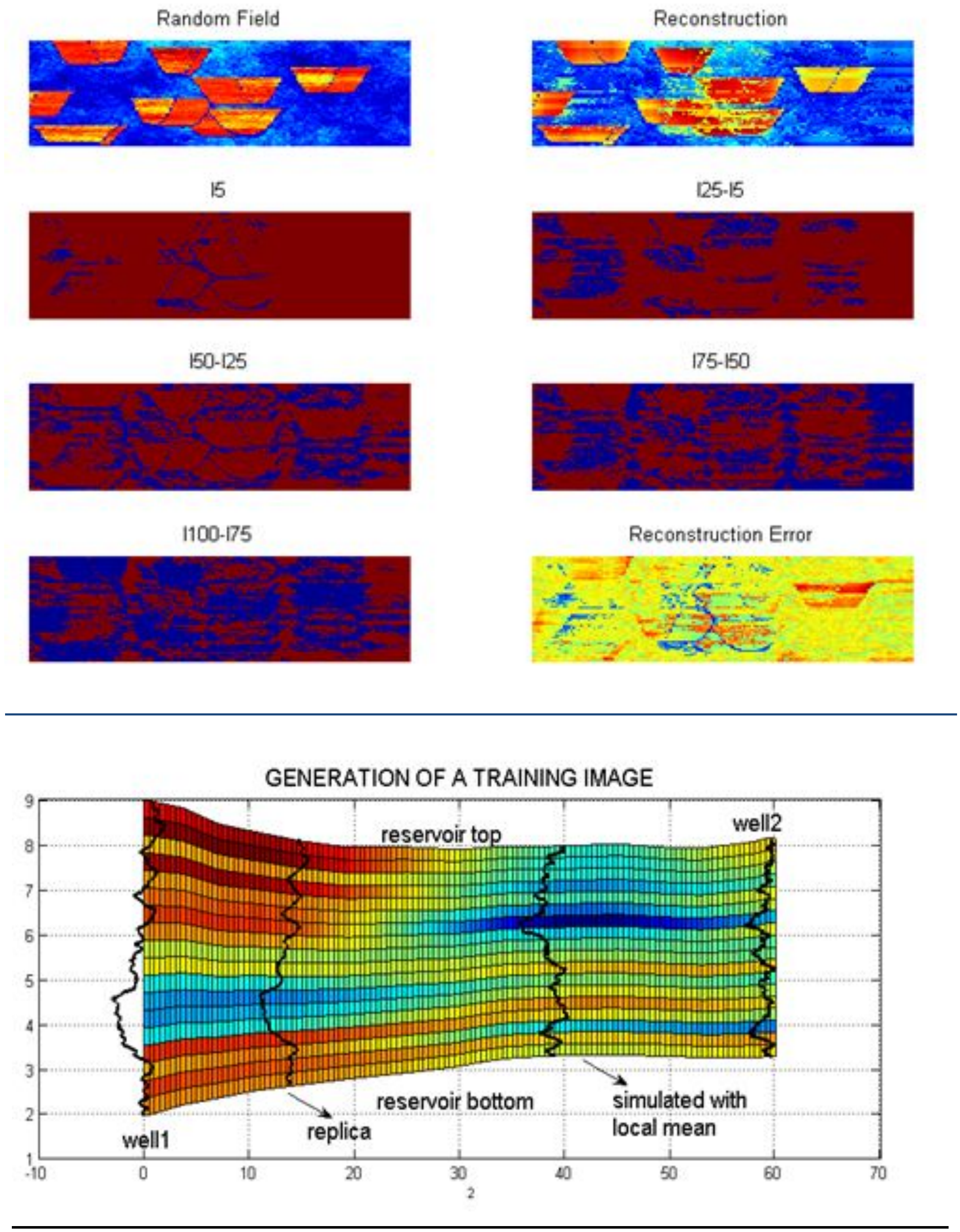

Figure 24: 2-d Interwell Projection.

The low frequencies of an image are taken from a 2-d seismic image which is reconstructed using the principal components of the image which are fit to well-log parameters. Plausible high frequency variations are added using a statistical model.

High frequency information, that is, the small scale variation in parameters, is made by fitting a statistical model to the well logs. A sample is taken from this distribution at a series of locations between the wells and stretched to match the structure. In this way, a plausible image is generated such as the bottom image of Figure 24.

Although it seemed promising, the problem with the method was that we could not find real data to support it. At Holly Branch, there is almost no seismic impedance contrast between layers in the pay zone which means that there is no correlated seismic image and this is not unusual in shale gas deposits. The only seismic attribute which showed contrast in the well logs was the Vs velocity. The fact that traditional seismic data is often not useful in the direct vicinity of shale gas, but that there might be a Vs contrast, strongly motivates the use of Amplitude Variation with Offset (AVO) which can be used to estimate a Vs section from surface seismic. [Aki and Richards, 1980].

The other issue with seismic-based parameter projection is that unless the seismic velocity profile is perfect, the depths of seismic layers may not be calibrated with depth in the well-log section. We experimented with a few methods to make this calibration, including doing the calibration by hand and stretching or compressing the section near the drill hole accordingly, interpolating the amount needed to stretch between the hole.

5.2.6 Simulation and Optimization Module Group

The Simulation and Optimization Module Group contains modules designed to synthesize all of the selected data as well as user inputs into determining the optimal location for a well and making physically realistic predictions of its performance.. If none of the previous data analysis modules has been run, then the simulation modules become pure simulation modules, returning production curves (and eventually pressure fields) for purely theoretical wells, the parameters for which are entered through the **Data Manager, Reservoir** tab.

5.2.6.1 Automatic Well Placement

Inputs from Data Manager:

- *Reservoir Parameters: Fracture direction*

Inputs from other modules:

- *Decline Curve Analysis: decline curve parameters from several wells*
- *Decline Curve Analysis: cumulative production*
- *Parameter Estimation: permeability and fracture half-lengths for wells*
- *Shut-In test Analysis or Pressure/Rate Model Based Analysis: estimates of permeability and fracture half-length for several wells*
- *Unit: grid containing net pay thickness between wells*

Outputs:

- *Grid containing EUR after 10 years for all of the selected wells*
- *Location of a horizontal well that maximizes EUR while still maintaining a distance farther than the estimated fracture half-length of nearby wells*
- *Set of parameters including porosity, net pay thickness, suggested horizontal well direction and length for a new, optimized well*

This module contains the most sophisticated optimization software in the current version of SeTES. The goal of the module is to integrate all available information into a suggestion for the best location for a new well and to estimate other parameters for the well, such as its porosity, net pay thickness, and permeability, in order to use the simulation modules to predict production with a high degree of accuracy.

The module itself is divided into two parts:

1. A statistical model based on a Bayesian Network, which uses all available information to find an estimate of Ultimate Recovery that has the highest possible probability.
2. An interpolation algorithm to interpolate between the EUR estimates, applying constraints based on the position and fracture characteristics of nearby wells, in order to find the best location for an infill well.

In the Decline Curve Analysis, Parameter Estimation module, SeTES gives an estimate of ultimate recovery by simply projecting a fitted curve out 10 or 15 years and summing production (this is standard industry practice). Although a curve fit in this way gives an answer which is a good starting point for measuring the relative production of wells, the actual number that is produced should be treated with suspicion. The same data can be fit equally well using any number of sets of parameters, which, when projected out a long way in the future, give widely different estimates of production. If just a single answer is given for EUR, the algorithm used to find the parameters of a decline curve might have a large effect on the answers, with some algorithms producing ultimate recovery numbers that are as big as a factor of two or more over others. This is discussed at length in the Results section of this report under Decline Curve Parameter Fitting.

To improve EUR estimates, the first part of the Automatic Well Placement module (which might be moved to another location under Production Analysis in a subsequent version of the system) uses a statistical model known as a Bayesian Network to incorporate as much knowledge and independent assessments of parameters into the EUR estimation.

A Bayesian Network is a directed a-cyclical graphical model which can be used to compute the probability of a variable based a prior probability distribute and its conditional dependence on other variables. They are well described in Russel and Norvig (2003). All variables in the network are considered to be stochastic, that is, they do not take on just one value, but represent random samples from a probability distribution. The arrows between nodes denote dependence that is defined either by a deterministic relationship (such as a formula like Equations 2 or 3) or conditional dependence. In a Bayesian Network (as opposed to other types of probabilistic graphical models) all of the lines have arrows, meaning that the relationship between one variable and another is only computed in one direction. Furthermore, there are no loops—once the probability of a variable being in a given state is computed, the network has no further effect on that variable. Furthermore, the basic definition of conditional independence states that, no matter how many links are in the graph, the value of a variable in a node is dependent only on the values of the nodes that point directly to it (its "parents") and not the nodes which point to the parents themselves. In this way, a node is conditionally independent of the rest of the graph given its parents.

Bayesian networks and techniques for using them are extensively discussed in literature on computer learning such as the classic reference by Russell and Norvig [2003]. A considerable amount of research has been done in recent years to apply statistical models such as Bayesian networks and influence diagrams to problems in petroleum engineering [*Rasheva et al.,* 2011; *Ghoraishy et al*, 1998; *Al-Yami et* al, 2011]. Rajaieyamchee and Bratvoild [2009] built a Bayesian network to help support drilling decisions. Al-Yami et al. [2010] built a small-scale Bayesian network-based expert system for the optimal design and execution of successful cementing practices. Liu et al. [2011] suggested a Bayesian network early fault detection for rod pump systems.

There is no magic in a Bayesian Network. It is simply a model used for sampling values and computing probability distributions of variables that have dependencies that are too complicated to write out conveniently. It is much like of a bookkeeping tool. Because the probability distribution of a value in a node depends only on the values of its parents, efficient algorithms can be designed to propagate probabilities up and down the graph. In other words, if the values of a few of the nodes are known (or have been measured), then it is possible to compute the distribution of all of the other nodes in the graph below given the known nodes.

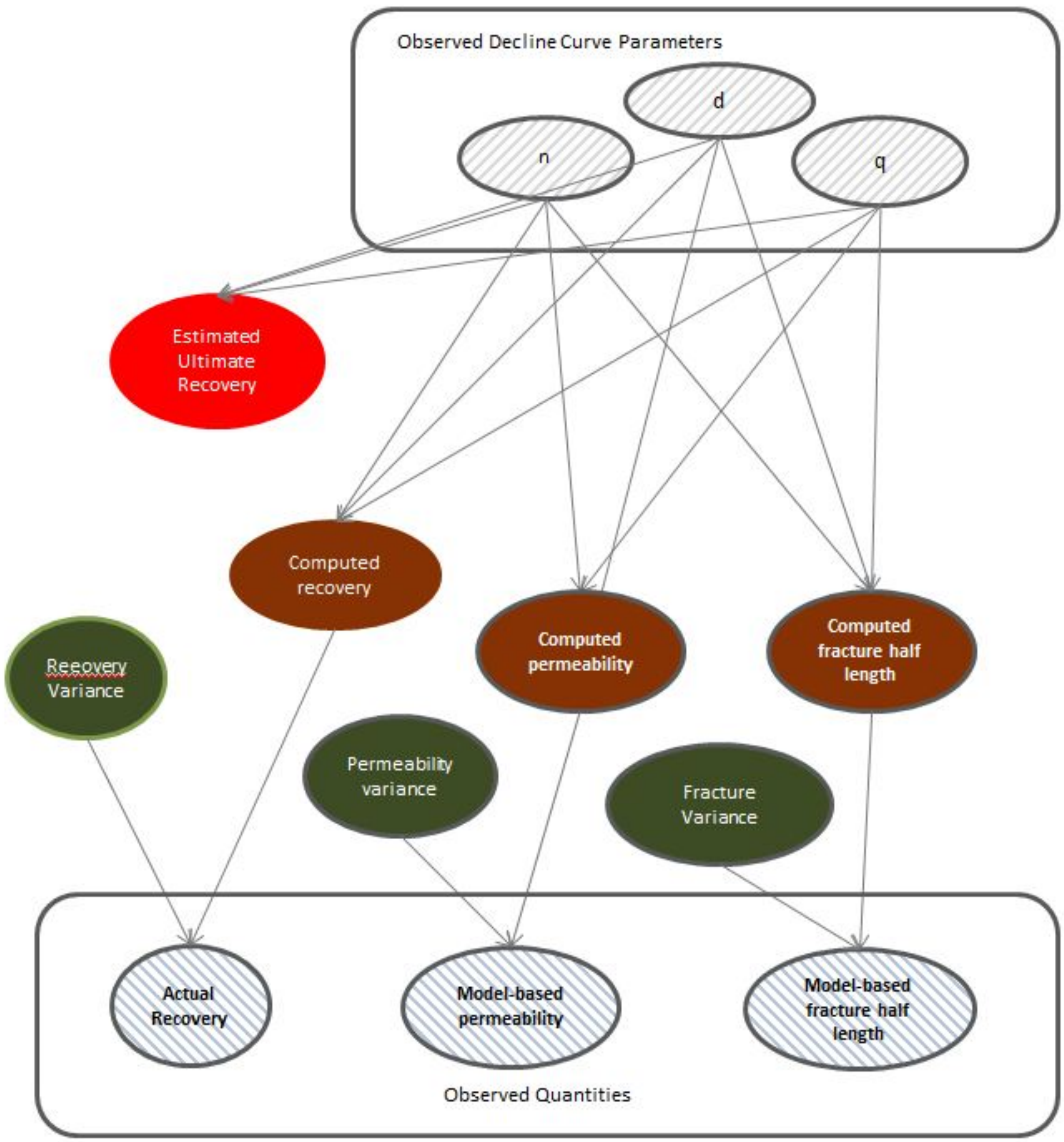

Figure 25: Bayesian network for estimating Production.

This network encodes conditional dependency between the variables in the nodes. The striped nodes are variables that have values that can be observed. The brown quantities can be directly computed using a formula. The red value, EUR. is the node that the network is designed to infer.

and y, are linked as in Figure 20, then the probability of y given x is known. Bayes' Rule states that

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} \qquad 5.$$

P(*x*|*y*) is known as the posterior distribution of *x*, *p*(*y*|*x*) is called the *likelihood* and *p*(*x*) is known as the *prior*. Intuitively, the posterior represents the distribution of values in an unknown node. The *likelihood* is the conditional distribution given on the link between *y* and *x*; the *prior* represents one's belief about what would be reasonable values for *x* and *p*(*y*) (sometimes called the "evidence"), where *p*(*y*) is the normalizing constant required to make sure that the posterior distribution adds up to one.

In practice, a network is set up by identifying all of the relevant variables in a problem, assigning unknown values prior distributions and trying to make the conditional values in the links as realistic as possible. Often it is useful to introduce "deterministic" nodes, that is, nodes that are a direct function of other variables and not *per se* random variables in their own right. Once the network is set up, an algorithm called *Markov Chain Monte Carlo* (MCMC) is used to take random snapshots of values of variables in the nodes according to the network's distributions. Without going into the details of MCMC, any state of the network (that is, a set of values in each of the nodes) is considered a "sample". The algorithm starts with a random guess and then assesses the probability of that guess being a realistic state, thereby deciding if it is going to keep the sample. It then adjusts some of its internal parameters that tell it how to make a sample and reiterates, keeping a record of all of the samples along the way. After a burn-in period, the algorithm converges to a set of rules for sampling that ensure that the samples follow the true distribution of the network. After a sufficiently large number of samples have been taken, it is possible to see what the real distributions are for the various nodes by making a histogram of the samples after the burn-in period samples are discarded[2].

The Bayesian Network SeTES uses for EUR estimation is shown in Figure 25.. The fact that the lines are arrows, not simple lines, denotes that the relationship between nodes is only computed in one direction – that is, we model the dependence of Computed permeability

---

[2] There tends to be confusion in the earth science community over the point of MCMC sampling. We are used to the idea of ill-posed and under-determined problems and tend to think in terms of a solution space, that is, mapping out the infinite answers to a problem, or restricting solutions to a particular part of the space that makes physical sense. The goal of MCMC is not to describe a solution space, but to sample answers from that space in accord with the probability that they ought to occur if you kept solving equivalent variants of the problem. If the goal is to map the solution space to see what answers might be, then MCMC is not an efficient algorithm because, in the course of its iterations, it revisits the same solutions over and over, particularly high-probability solutions.

on decline curve parameters and not the other way around. The striped nodes indicate variables or parameters that can be directly observed. An astute reader will recognize that the decline curve parameters and permeability and fracture half-length are not observable parameters as they have to be inferred from data. In a complete model, this inference would be built into the graph by making them dependent of decline rate and pressure data; however, the amount of time it would take to compute estimates of other parameters in the graph using the full model would be prohibitive. We cheat by assuming that the parameters have been "measured" but are uncertain with a given mean and variance. The brown nodes are quantities that can be computed directly from other nodes using a formula (such as equations 2 and 3). Priors on all of the parameters are taken from the probability distributions stored in the SeTES database.

The goal of the network is to find the EUR estimates that have the highest probability (Maximum a Posteriori [MAP] estimates) given their parameters from decline curve analysis, parameters from a shut-in test or rate-pressure analysis, and actual production of the well. We use a freeware Python package called pymc (Patil et al 2010) that makes it simple to set up and test models and which handles the MCMC sampling via a set of embedded C programs. The module supports both MAP estimation of EUR as well as a full posterior sampling of EUR. Currently only the MAP estimate is displayed because computing the full posterior for each well can take up to half an hour. (This is why we are considering splitting the module and moving the EUR estimation into Production Analysis so that the user could be doing other things while the full distribution is being computed).

Once a MAP estimate of EUR is computed, the standard interpolation subroutine is used to make a grid of EUR values between the wells. SeTES chooses the point with the highest EUR that is outside of the range of the fractures of the other wells as the place to make its suggestion for a new well. It then looks at the results of the Unit module to determine a probable thickness at that location, at the well-log module to come up with porosity, and at

the parameter estimation module to estimate permeability (very shortly, this estimate of permeability will be replaced by a MAP estimate from the network). If a direction of fracturing has been entered, SeTES will use it to compute the optimum direction for a horizontal well.

The results are displayed currently as a surface plot showing the MAP estimate of EUR and the location of the new well with respect to existing wells as shown in Figure 26 . The location of the well will be (and was in an earlier version of the system) displayed as an interactive image. The user can move the location of the new well SeTES will recompute parameters and display an estimated EUR accordingly

One of the great advantages of using a graphical model such as a Bayesian Network to represent relationships between data is its flexibility. Once the model has been put together, it is easy to add new nodes representing new types of data. Although most of the other modules are fixed and will not change dramatically other than to add features, the Automatic Well Placement module is expected to grow as the SeTES team gets more experience with different types of data. The python "Expert" which runs the module is written so that new nodes can be written into the network and an appropriate model chosen depending on the data that the user has selected.

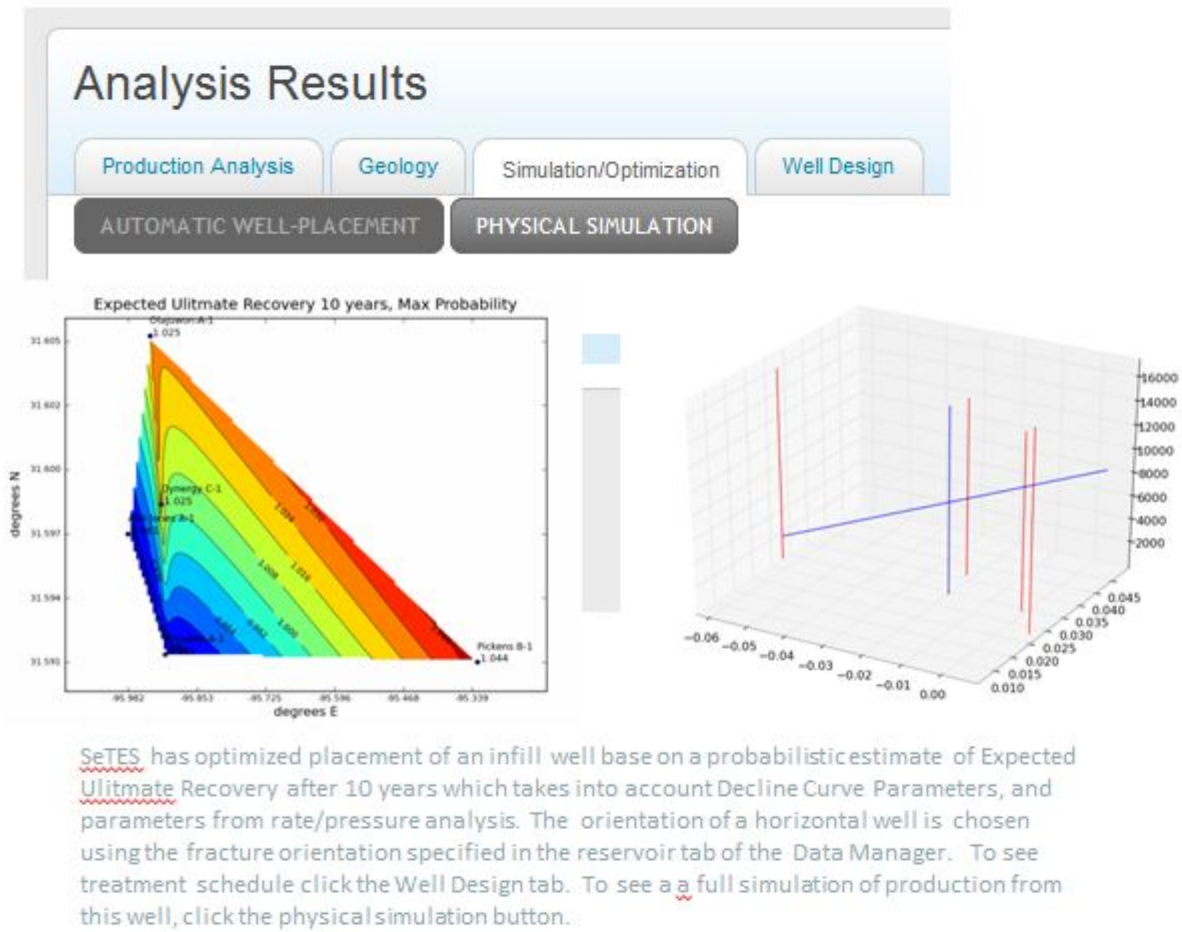

Figure 26: Automatic Well Placement

5.2.6.2 Physical Simulation

Inputs from Data Manager:

- *All Reservoir Parameters that have been entered*

Inputs from other modules:

- *Automatic Well Location: permeability, porosity, net pay thickness for suggested well location*

Outputs:

- *Production profile for simulated well*
- *Pressure field around well*

The Physical Simulation module takes the outputs from the Automatic Well Placement module and uses a 3-D modeling code to predict the production from a theoretical well designed with those parameters. If parameters from the Automatic Well Placement module have not been computed, they are taken from the data manager Reservoir Parameters section. If no parameters have been entered, they are taken from default values in the SeTES database. The parameters that are used are displayed in an interactive form to the right of the output figure showing predicted production along with the module that was used to derive them. The parameters can be updated and the simulation will be re-run, however it takes several minutes.

The ShaleSIM modeling program underlying this module is a member of the TOUGH simulator developed at LBNL. It is implemented in Fortran for computational efficiency and linked to SeTES through a python execution call.

The TOUGH family of codes provides multi-dimensional numerical models for simulating the coupled transport of water, vapor, non-condensable gas and heat in porous and fractured subsurface media. These models describe the processes and interactions involved in the flow of fluids in the subsurface, including the appearance and disappearance of liquid and vapor phases, point and condensation, multiphase flow due to pressure, gravity and capillary forces, vapor adsorption with vapor pressure lowering, heat conduction, and heat exchange between rocks and fluids. Additional information is available in [*Pruess*, 1991, 1995; *Pruess et al*., 1996; *Wu et al*., 1996] . A full report on TOUGH is given in [*Moridis et al.*, 1998].

The TOUGH adaptation for SeTES returns both a production curve and the pressure profile for a horizontal or vertical well. Currently only the production curve is being displayed in the module tab, although early versions displayed only the pressure profile as shown in Figure 22.

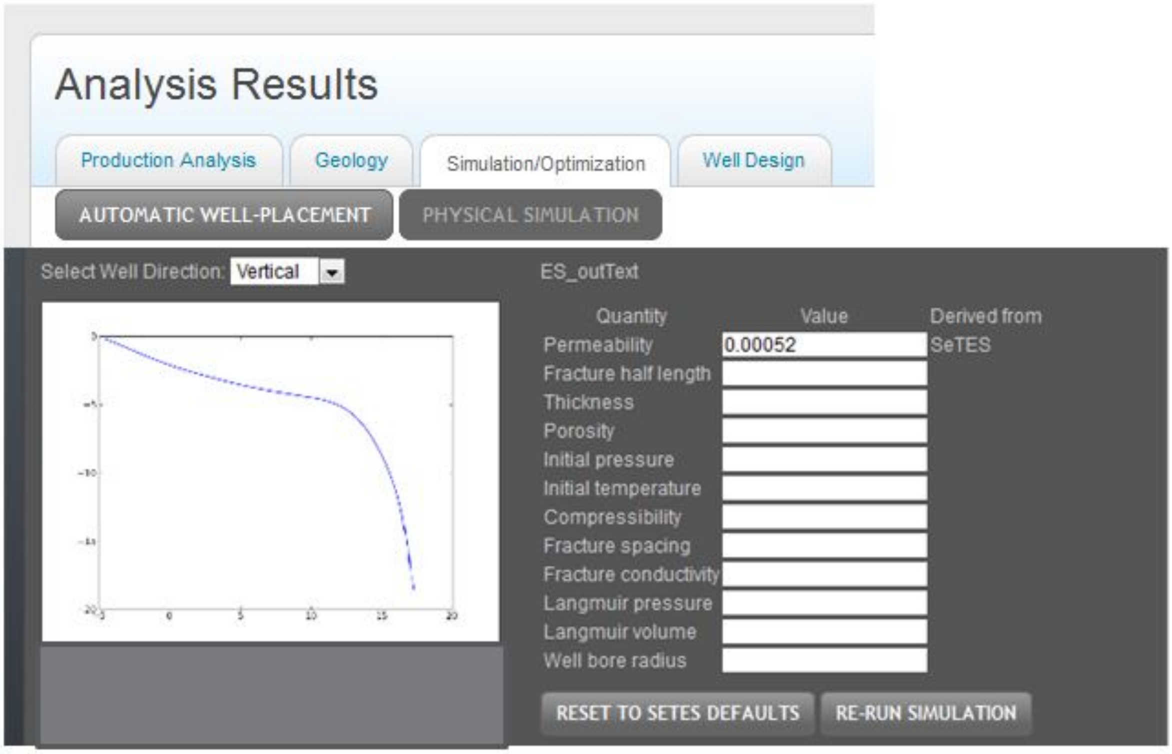

Figure 27: Physical Simulation module.

5.2.7 Module under Development for Simulation and Optimization

5.2.7.1 Interactive fracture-based simulation

Inputs from User

- *Fracture type, length and orientation chosen from an Interactive Fracture Visual Interface*

Inputs from Data Manager:

- *All Reservoir Parameters that have been entered and*

Inputs from other modules:

- *Automatic Well Location: permeability, porosity, net pay thickness for suggested well location*

Outputs:

- *Production profile for simulated well*
- *Pressure field around well*

This module will contain the most sophisticated modeling tool in SeTES. It is a finite-difference modeling code implemented in Fortran 95 which has the capability not only to model general fracture spacing and length, but specific fracture patterns. The program behind it, is a numerical simulator which models multi-component, multi-phase, non-isothermal flow in fractured media, using a very fine grid to accurately describe fluid flow [*Freeman et al*., 2010].

The fracture patterns can be input into the system by means of an Interactive Visual Fracture Interface (IVFI), which was designed to make model setup more intuitive that the traditional use of text-based input files (Figure 28). The IVFI was designed and fully implemented in PHP for the first prototype of SeTES, but is waiting for full implementation in the current system. By default, the system chooses the first fracture type and uses a

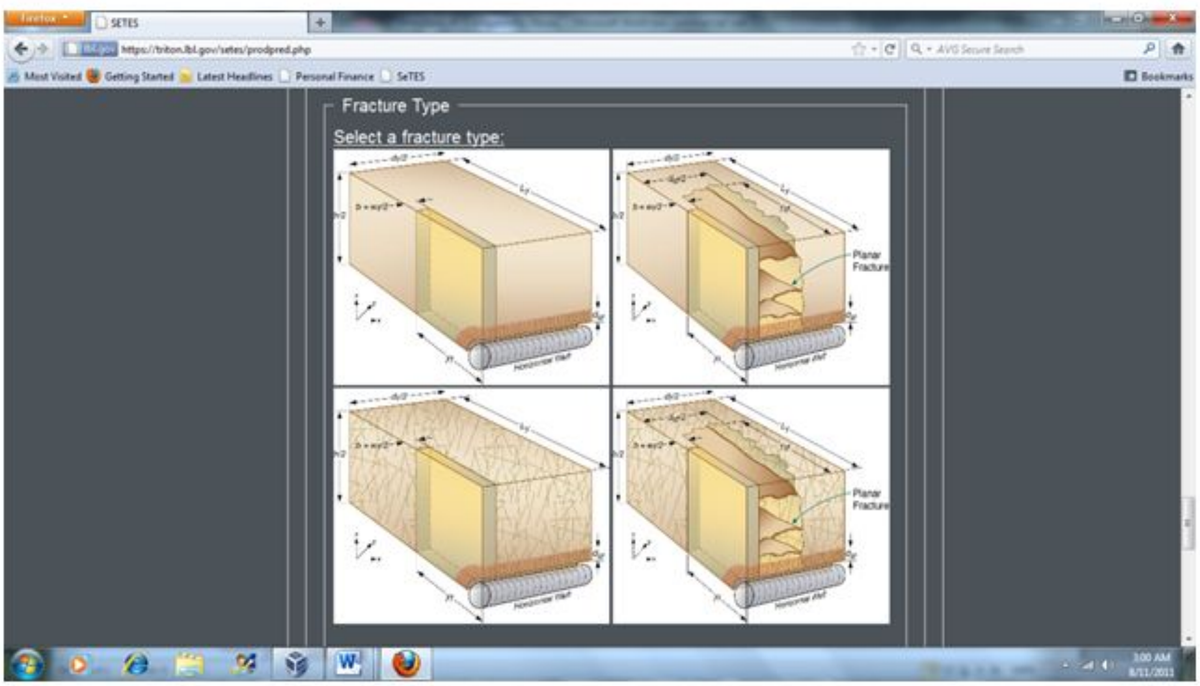

Figure 28: InteractiveVisual Fracture Interface.

fracture half-length from the Automatic Well Location module. Unlike the other modules, however, this module will wait for the user to agree to the fracture selection before it runs automatically. This is because the module is computational expensive and the run time can be quite long.

Because the simulation grid is very large (involving regularly 100,000s of gridblocks) in order to capture fine-scale interactions the compute time for this module can be extremely long, up to half an hour or more which is too long for a web-browser to wait. In the results and discussion section, we discuss how we might go about integrating it into the system without crashing the server or causing the user's web browser to time out and lose the SeTES session.

As of this writing, the IVFI and the underlying modeling software are complete, but neither are integrated into the system. (The IVFI was fully implemented in a very early version of SeTES). The final implementation will require the development of the queuing system to allow the simulation to run in the background, allowing the user to perform such intensive analysis offline, with results available upon the user's return.

5.2.8 Well Stimulation and Treatment Module Group

The Well Stimulation and Treatment group includes modules which suggest final production schemes for a well which has the properties of the suggested well output from the AWP module. It has no modules which estimate petrophysical parameters.

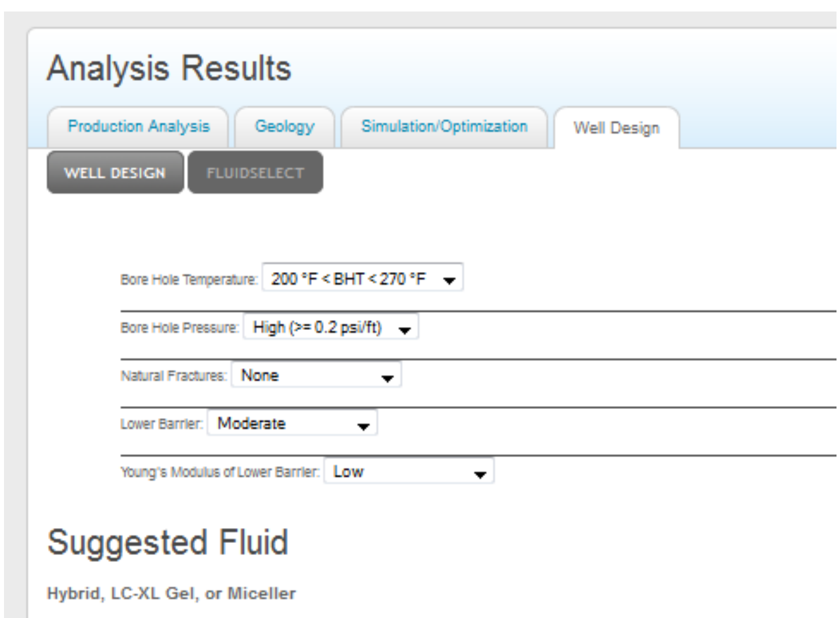

Figure 29: Fluid Selection Tool.

5.2.8.1 Fracture fluid selection

Inputs from User

- *Parameters of the well input through an Interactive Menu*

Inputs from Data Manager:

- *Any parameters that have already been entered*

Outputs:

- *Suggested hydro-fracking fluid*

The Hydro-fracking Fluid selection module is based on a decision tree, the base of which is given by Holditch [1993]. The tree itself is implemented as a set of questions generated in PHP and displayed as a self-updating cascade of forms, as shown in Figure 29. For example, the first question is “Bore Hole Temperature” with selections, “<200 F”, “200<BHT<270 F”, and “>270 F”. If the first option is selected, then another menu asks for Bore Hole Pressure, giving options “Low (<0.2 psi/ft)” and “High (>0.2 psi/ft)”. If the first option, Low, is selected, the system suggests a fracturing fluid “N_2 foam”. If the second option is selected, then a new menu asks about Natural Fractures (‘None” or “Many”) and so on. The full logic of the tree is shown in Figure 30. Currently, the selections must be made by the user, however in the next version, SeTES will attempt to automatically fill in as many of the field as possible using all available data.

5.2.9 Modules under Development for Well Stimulation and Treatment

Well Simulation and treatment is the least developed module group of SeTES because the optimization modules described below were developed late in the game. They are, however, some of the simplest modules to implement and should be fully functional soon after delivery of the Alpha system.

5.2.9.1 Proppant schedule optimization

Inputs from User:

- *Proppant characteristics: permeability, specific gravity, porosity of proppand in closure*
- *Fluid data: an exponent-rheology, K-factor – rheology, spurt loss consideration, value of kappa constant*
- *Operational data: expected fracture height, injection rate for one wing, injection rate for two wings*

Inputs from Data Manager:

- *Young’s modulus*

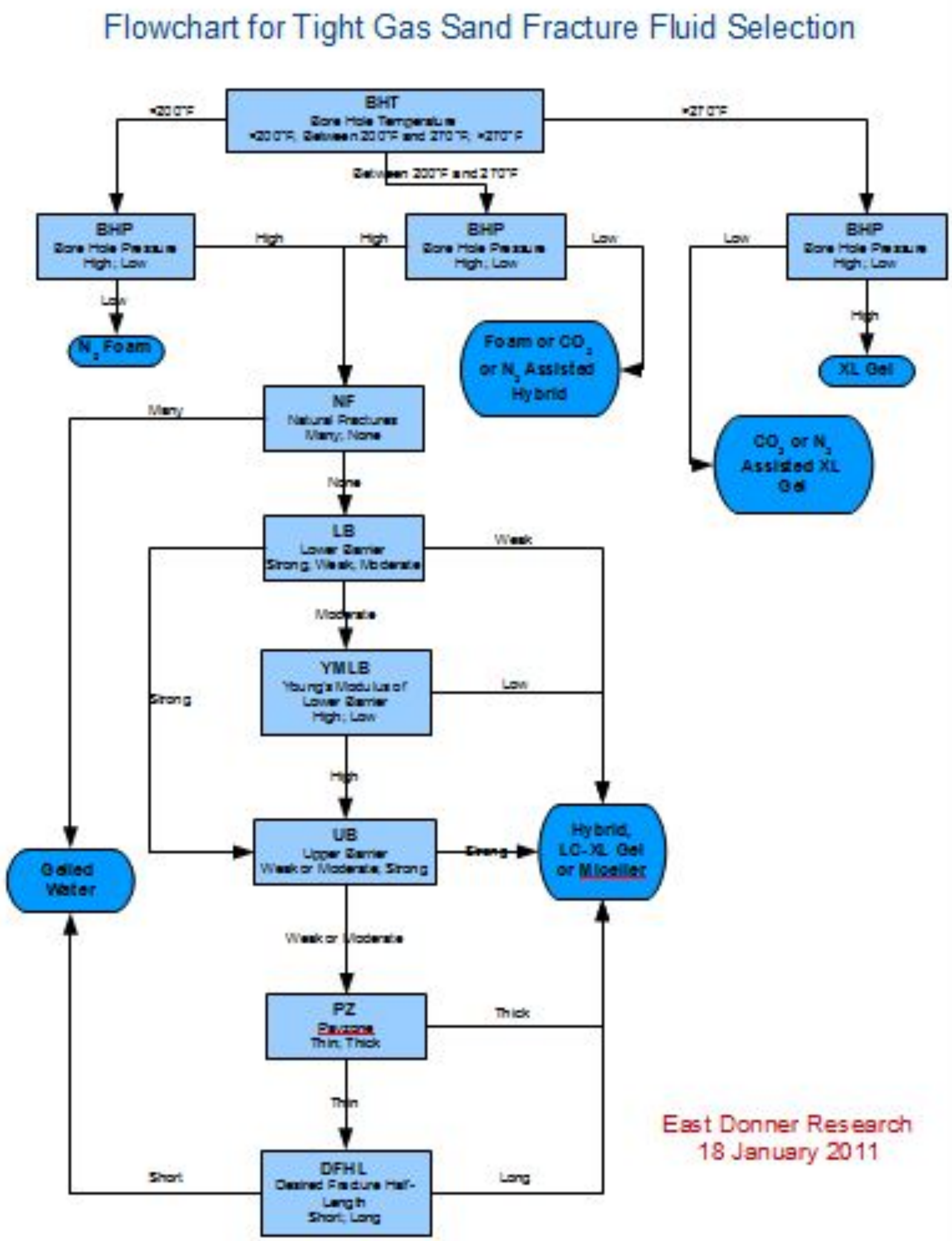

Figure 30: Logic behind fluid selection menu.

A slight modification of this is the Open Office-generated flow chart which can be directly compiled into PHP code using the graphical compiler described in section 5.4.5.2 Graphical Compiler.

- *Automatic Well Placement: permeability, net pay thickness, gross pay thickness, drainage size*

Outputs:

- *Target fracture length, fracture height, proppant width, proppant number, total injection time*

This module implements the Perkins-Kern-Nordgren method to estimate optimum fracture dimensions to achieve maximum productivity, and then returns the proppant schedule necessary to achieve those dimensions.

Currently the module is completely implemented as a Fortran 90 program. Since the programming itself is simple—mostly a decision tree with a small optimization routine—we are considering whether or not it would be easier to implement it directly in Python, as there is minimal need for numerical optimization. Samples of the Fortran input and output files are shown in Figure 31.

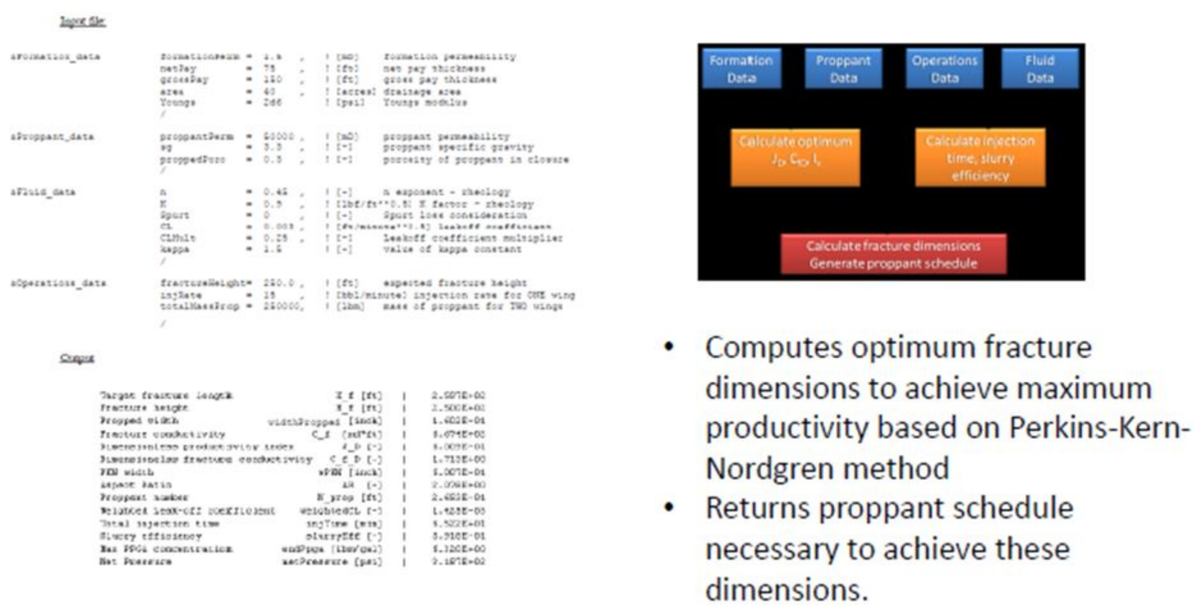

Figure 31 Proppant schedule optimization module.

5.2.9.2 Treatment Cost Optimization

Inputs:

- *Permeability, porosity, stress orientation, operational constraints and geometric constraints*

Outputs:

- *Horizontal well length, fracturing fluid viscosity, proppant concentration, injection rate of fracturing fluid and injection time*

This module optimizes Net Present Value of a horizontal well taking into account physical parameters of the well and operational costs. It uses the LMN model for hydraulic fracturing propagation [*Lietard et al*., 2007] with the optimization algorithm NLPQLY [*Schittkowski,* 2009]. It outputs suggested well length, fracturing fluid viscosity, proppant concentration, and injection rate and time. The Fortran for the module is complete and has been tested [*Bhattacharya*, 2011] but has not yet been integrated into SeTES.

5.2.10 Data Download, Session and Checkout Module

Short of right-clicking on figures and copying numbers by hand, there is currently no module for downloading data, results or figures from a SeTES-Alpha session. Furthermore, a user can only have one session (i.e. be working on one set of data, although the data can change or be expanded to during the session) and there is no mechanism in place to save the results of the session so that the user can go back and resume it after logging off. These are all issues that must be addressed in the Beta version of the system. The approach to solving these problems is highly dependent on discovering what users tend and want to do with SeTES. We will have a much better conception of what the appropriate procedure ought to be after we have the results of basic usability testing.

The features we currently believe to be important include:

- **Checkout.** A user needs to be able to download:
    - Figures
    - An automatically generated report
    - Text files containing fits to data, parameters interpolated between wells, or other results. This could exist at the module level, or as a site-wide “shopping cart.”
- **Save**. The user should be able to save the session and return to it
- **Exit**. Upon exit, data which was selected as “private” should be erased and the SeTES database updated with parameters and statistics computed during the session.

We have also discussed making certain parts of the code itself, such as the Python "experts" or a disk image of the whole system available to users, who could continue their work offline. The modules are largely independent and could be useful to researchers with suitable programming skills.

5.3 Uncertainty

In order for uncertainty to be propagated through the SeTES system, it first needs to be estimated. This estimation is done within the modules themselves, with different modules using different algorithms. The three algorithms which are implemented in the system, and discussed in their appropriate module section, are:

- **Jackknife**: a sampling method in which estimates of parameters are repeatedly made using different subsets of data; used in Decline Curve Analysis. The variations on estimations of parameters made using different subsets of the data are used to estimate a mean and variance for the parameters.
- **Cross-validation**: a similar method in which subsets of the data are held out, the rest of the data is used to estimate parameters and the difference between the predictions of the held out data and the actual data are used to estimate accuracy of the overall prediction; used in SeTES interpolation algorithms.
- **Statistical modeling**: explicit modeling of probability distributions and establishing of a posterior distribution over parameters using a graphical model and MCMC sampling; used in Automatic Well Placement.

5.3.1 Polynomial chaos

One additional algorithm was explored for use in SeTES and may be implemented in the Beta release: uncertainty quantification via Polynomial Chaos expansions. Although the method is powerful, we were not able to develop it far enough for use in SeTES Alpha. However, if implemented it in future versions, it will take the self-teaching capacity in a new and exciting direction.

Polynomial Chaos is a basis expansion, not unlike a Fourier Transform, where a set of realizations of a process, such as outcomes from a modeling program, are projected onto a basis space that is formed, not by deterministic functions such as sines and cosines, but into a space made up of orthogonal polynomials that are formed from the variations on the inputs. The technique has been implemented for applications in chemical and physical systems [*Reagan et al*, 2003; *Ghanem and Spanos*, 1990; Kuzma et al, 2011]. To our knowledge it has never been applied in petroleum engineering.

The fundamental concept behind a Fourier Transform is that most functions can be decomposed into a set of sines and cosines. Once the transformation has been made, it is possible to manipulate the contribution of the various sines and cosines to the function, bringing out details that may be of interest, suppressing noise and modeling relationships through techniques such as the Weiner Filter.

It is, of course, possible to expand functions using a set of basis functions that are not sines and cosines. In Polynomial Chaos (PC), a function of multiple variables is expanded onto a set of orthogonal Hermite polynomial functions of those variables. Instead of manipulating the data directly, parameters of the orthogonal functions can provide a compact representation of the data. Among other features, a PC expansion makes it simple to determine the importance of not just the uncertainty in individual parameters, but the effect of uncertainty in combinations of parameters for a complicated dataset, such as the output from a modeling program.

Consider a relationship that can be written in the form:

$$\vec{d} = G(\vec{m}) \qquad 6.$$

Where *d* is a vector containing measured or modeled data and *m* is a vector containing parameters of interest. *G* is the output of a modeling program that can be used to compute new data from parameters *m*. For example *m* could be a set of Blasingame/Ilk decline curve parameters and *d* could be a production curve computed using Equation 1.

A Polynomial Chaos expansion for *G* is given by:

$$G(\vec{m}) = \sum_{i=1}^{\infty} \lambda_i \Psi_i(\vec{\xi}) \qquad 7.$$

where the $\Psi_i(\vec{\xi})$ are a set of orthogonal polynomials which make up a basis for *G*.

The major difference between a Fourier Transform and PC is that in Fourier, the transform variables (time and frequency) are well-behaved deterministic variables. $\vec{\xi}$, however, is thought of as a vector of *stochastic* variables which are linked to variations in $\vec{m}$. Generally $\vec{m}$ is a function of $\vec{\xi}$ such as

$$\vec{m}\left(\vec{\xi}\right) = \vec{m}_0 + \sigma\vec{\xi} \qquad 9.$$

The exact form of the polynomials depends on the distribution of $\vec{\xi}$. If it is a normal distribution, then the correct polynomials are the Hermite polynomials.

The coefficients of a PC expansion, λ_i are computed by projecting G (as a function of $\vec{m}$, which is itself a function of $\vec{\xi}$) onto the polynomials of the expansion using the formula:

$$\lambda_i = \frac{\left\langle G\left(m\left(\vec{\xi}\right)\right)\Psi_j\left(\vec{\xi}\right)\right\rangle}{\left\langle \Psi_i\left(\vec{\xi}\right)\Psi_i\left(\vec{\xi}\right)\right\rangle} \qquad 10.$$

The point of the expansion is that the function *G* has now been expanded in a space that is defined by variations in its input parameters. This means, among other things, that the contribution to the variance in *G* due to a particular parameter, or combination of parameters can be computed by "filtering out" or setting to zero all of the λ_i which are not directly associated with that parameter or combination. It also makes for a convenient representation of what might otherwise be complicated distributions, because instead of storing raw data, histograms, or statistical metaparameters, distributions can be stored in terms of their projections into PC space, which are sufficiently described with just a few coefficients of the expansion. (In SeTES, we have worked with the concept of storing sets of PC coefficients in the database in place of the conventional mean and standard deviations). Finally, a set of simple functions of the PC coefficients can be used to propagate uncertainty through functions of the underlying distributions, as we have already derived a linkable setoff "overloaded operators" that allow mathematical operations on the coefficients of the expansion—representing operations on the full distribution of each variable.

PC expansions are a very attractive means of propagating uncertainty through expensive modeling programs because, unlike jackknife, cross-validation and Bayesian networks, the PC coefficients are a function of prior distributions on parameters in local areas—which means that they can be pre-computed. During the SeTES downtime, when it is not serving users, the system could be computing new realizations via the expensive modeling programs (such as ShaleSIM) and using them to pre-generate PC coefficients. That way, estimates of the uncertainty of input parameters could be included in the public database to guide future analyses.

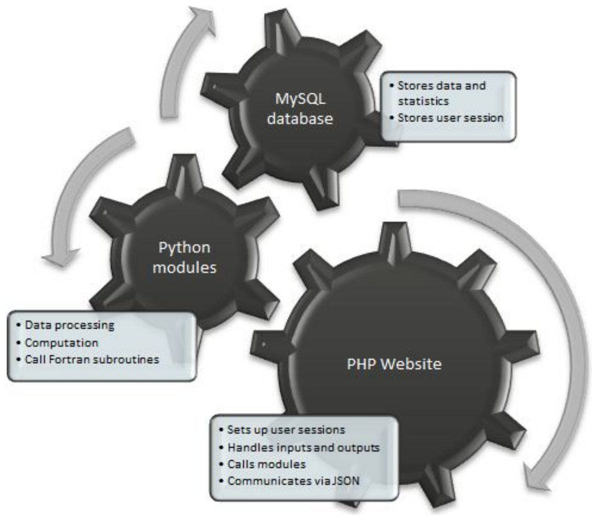

Figure 32: The Three cogs of SeTES.

5.4 SeTES technical implementation

As stated in the introduction, SeTES is composed of four basic units: a database; a set of modules which were described in previous sections; an "Expert System," which is not a separate program, but intelligence and decision-making capability built into the modules and linked through the database; and, finally, a user-friendly web-based interface. In this section we present a general guide to the three big pieces of the system, showing how they fit together to achieve a seamless experience for the user. Complete technical documentation for SeTES developers and webmasters is included in Appendix I.

Figure 32 shows the three major cogs of SeTES which have to turn in harmony: the website, the modules and the database.

5.4.1 Website

The web interface for SeTES is implemented in a customized PHP-based framework called Symfony, which is responsible for producing the initial web page views seen by the user. HTML provides the basic structure and layout of the website, supplemented by a JavaScript framework called jQuery that provides the interactive features, including the launching of individual computational modules. When the user triggers a request to perform an analysis or database search by clicking on one of the tabs or buttons on the website, the PHP code uses a module called an analysis runner to gather the input data and send a Javascript Object Notation (JSON) data structure to the expert system Python code, telling it to

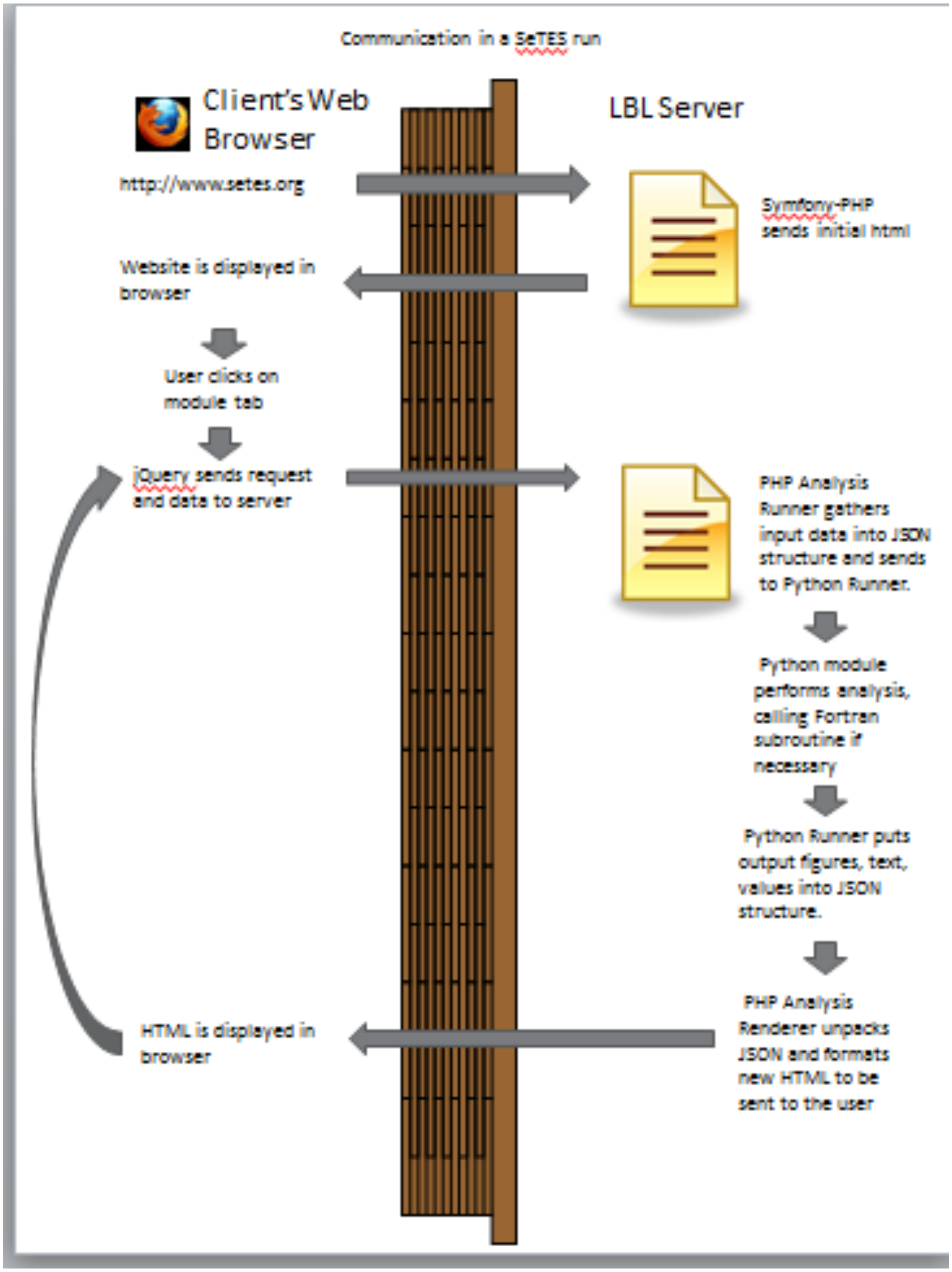

Figure 33: Illustration of a SeTES run.

execute the desired module with the input data. At the top of all of the python modules sits a master program called the Python Runner. If the user is just logging on to the website, the Python Runner sets up a "session," which is the data structure that will hold all of the information about the session, i.e. variables that have been assigned values and data that have been selected. It then parses the JSON and runs the appropriate modules, which, if necessary, call relevant FORTRAN programs. At the end of the run, the Python Runner collects all of the resulting data and formats it as another JSON data structure, organizing the computational results into categories that can be understood by the PHP. The PHP then receives the output data (which can include text, numbers and figures) and invokes a module called an Analysis Renderer to unpack the JSON data structure and format the output via new HTML, which is sent to the user and seen in his browser. Although the user has the illusion of SeTES existing in his web browser, this is not the case. All of the computation during a session is done on the server; all the user receives are static displays. This is why when new values are entered into boxes in modules such as Decline Curve Analysis the corresponding images do not update unless the user pushes a re-run button. A schematic for all of this communication is given in Figure 33.

There is a huge difference between a website and other interactive environments such as MATLAB, EXCEL, word processors, etc. which sit on a user's own computer. *A website has no memory.* When the user goes to a webpage by typing an address (such as www.SeTES.org into the browser, his machine sends a request to the server through an electronic pipe (which is set up using a protocol called TCP) which includes the address of the server, a specific file perhaps, and some data or requests. The server accepts the request and displays an HTML document, or in the case of SeTES analysis, runs a PHP program that outputs an HTML file that is sent back to the client. The user may decide to fill out a form, such as the forms on the Decline Curve Analysis module, and send in another request carrying those values, but the idea that the decline curve parameters actually exist in an Expert System on the user's machine is an illusion. In reality, they are saved on the server side only, and the web browser is merely a means of communication with the remote expert system.

A *state machine* is a computer program that computes values or actions dependent on the state of a set of variables in the system. Given new variables, it goes into to a new state, computes again, and so on. The SeTES expert system was envisioned very much as a state machine, making decisions based on the state of the system at every step. A web program, however, is not a state machine from the user's side—functionality is generally limited to sending requests and received HTML documents in return. Many of the biggest programming challenges in SeTES have been to come up with structures and protocols to handle the passing of information from the user to a simulated "state" on the server side and in and out through the modules, giving the user the illusion of a seamless system. If we had started development in a simpler environment, one with local memory and global variables, designing the system would have seemed easier. On the other hand, problems of subsequently implementing it on the web would have been enormous, and success would not have been assured.

5.4.2 Database

A screenshot for the MySQL database supporting SeTES is shown in Figure 34. This database sits on the LBNL server and can be conveniently access through a utility called PHPmySQL. Through this utility, data, fields, and tables can be added directly to the database by a user with proper administrative privileges.

localhost ▸ setes_v2_test

Structure | SQL | Search | Query | Export | Import | Operations | Privileges | Drop

Table ▲	Action	Records [1]	Type	Collation	Size	Overhead
application		5	InnoDB	latin1_swedish_ci	16.0 KiB	–
application_file_type_limits		5	InnoDB	latin1_swedish_ci	32.0 KiB	–
file		30	InnoDB	latin1_swedish_ci	96.0 KiB	–
file_credential		30	InnoDB	latin1_swedish_ci	32.0 KiB	–
file_property		0	InnoDB	latin1_swedish_ci	16.0 KiB	–
file_set		2	InnoDB	latin1_swedish_ci	32.0 KiB	–
file_set_member		17	InnoDB	latin1_swedish_ci	32.0 KiB	–
file_type		14	InnoDB	latin1_swedish_ci	16.0 KiB	–
formation		6	InnoDB	utf8_general_ci	64.0 KiB	–
group		1	InnoDB	latin1_swedish_ci	32.0 KiB	–
invite		0	InnoDB	latin1_swedish_ci	64.0 KiB	–
reservoir		6	InnoDB	latin1_swedish_ci	64.0 KiB	–
reservoir_formation		6	InnoDB	latin1_swedish_ci	32.0 KiB	–
reservoir_property		18	InnoDB	latin1_swedish_ci	16.0 KiB	–
role		2	InnoDB	latin1_swedish_ci	16.0 KiB	–
unit		12	InnoDB	utf8_general_ci	16.0 KiB	–
unit_property		0	InnoDB	latin1_swedish_ci	16.0 KiB	–
unit_reservoir		12	InnoDB	latin1_swedish_ci	32.0 KiB	–
user		6	InnoDB	utf8_general_ci	48.0 KiB	–
user_group		1	InnoDB	latin1_swedish_ci	32.0 KiB	–
user_parameters		0	InnoDB	latin1_swedish_ci	16.0 KiB	–
user_role		3	InnoDB	latin1_swedish_ci	32.0 KiB	–
well		14	InnoDB	latin1_swedish_ci	64.0 KiB	–
well_property		18	InnoDB	utf8_bin	16.0 KiB	–
well_reservoir		14	InnoDB	latin1_swedish_ci	32.0 KiB	–
well_unit		21	InnoDB	latin1_swedish_ci	32.0 KiB	–
26 table(s)	Sum	243	MyISAM	utf8_general_ci	896.0 KiB	0 B

Check All / Uncheck All   With selected:

Figure 34: SeTES database tables.

The database as shown contains 26 tables. The tables user, user_group, user_parameters and user_role currently only contain the name and email addresses of registered users but are set up to contain much more, including information and results from previous sessions. The application table contains information about the modules, including required inputs, references, and text. The file, file_credential, file_set, file_set_member, and file_type tables contain information about data in the database that is stored in files such as production information and well-logs.

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

reservoir_id	name	value	mean_value	std_value
4	HeidiTest	20	1	0
4	formation_name	Bossier	1	0
4	frac_direction	25.632	30	90
4	fracture_conductivity	1.66	1.66	0.05
4	fracture_halflength	165	150	35
4	fracture_orientation	69	1	0
4	fracture_spacing	60	60	10
4	fractured_res	1	1	0
4	init_res_pressure	4860	1	0
4	init_res_temp	165	1	0
4	lang_press	0	1	0
4	lang_volume	0	1	0
4	mat_compress	0.000001	1	0
4	mat_poros	0.01728	1	0
4	net_pay_thickness	115	160	37
4	res_ thickness	1000	1	0
4	res_thickness	1000	1	0
4	reservoir_name	Holly Branch	1	0

Check All / Uncheck All *With selected:*

Figure 35: Reservoir table.

The basic data categories in the database are:

- **Formation**: a large-scale formation such as the Bossier or Marcellus shale
- **Reservoir**: a commercial reservoir or grouping of wells into a formation such as the Holly Branch Field
- **Well**: data pertaining to a particular well
- **Unit**: particular named layer within a formation, such as the Bossier shale that is part of the Bossier Formation

The reservoir_ property and unit_property tables have fields that contain statistics (mean and standard deviation) and default values for various properties such as porosity, permeability, fracture_orientation, and thickness. The reservoir_property table is shown in Figure 35. The field value indicates the default_value. The modules can query these statistics and use them as initial guesses for parameters. Similarly, the well_property table contains values that have been pre-computed in previous sessions for various wells.

When a file or other data is uploaded by the user, and file type "private" or "semi-private" is selected, then that information is added to the database only for the duration of the session. At the time of checkout, a query is performed and all of the private and semi-private records are removed. The mechanism for doing this exists in the Alpha-version, but is not currently used because there is currently no system-wide checkout procedure.

5.4.3 Example of a module

The general python structure of a module is illustrated in the following text. The name of the python file is Expert[module_name]. Here we are examining ExpertPCA, the Principal Component Analysis module, which depends on outputs from the Decline Curve Analysis Module. Before an algorithm becomes a module, it is first designed and tested as a free-standing environment. For example, many of the modules were first implemented in Matlab. One the general idea of the module is proven, it can be rewritten in Python using the SeTES development framework.

(The initial lines of the module are import statements that are not reproduced here)

The first line sets up a Python class for the module:

```
class ExpertPCA(ExpertBase):
    """
    Expert call 'PCA'
    """

    # @override
    _requiredInputs = ['BI_outputs', 'BI_names']

    # @override
```

```
_dependsOn = 'BIDecline'

    def run(self):
        vars = self.extractRequiredInputs()
        ....


        self._session.addOutputFile('figure', PCA_interp_uri)
        self._session.addOutputParam('PCA_outText', PCA_outText)
        self._session.addOutputParam('suspectText', suspectText)
```

It is followed by a list of inputs from other modules that it requires, and the name of any module which can be used to generate those inputs should they not already be present in the session. If the inputs do not exist, the dependsOn module will be run automatically. Finally, a set of methods (or subroutines) are defined. The first one is usually "run" which defines what the module will do when it is run, including how it will load data and what it will do given different types of data and checking to see if the module is in "re-run" mode. At the end of the module, a set of session methods writes out information to the session which will be translated into JSON and handed to PHP for output to the user.

It is also possible to save parameters as usr_session variables. The user session is the set of variables and values that are saved to be passed between modules. The appropriate syntax for setting a user variable is:

```
self.saveParameters({ 'BI_outputs': decline_outputs, 'BI_names': well_names })
```

This example was pulled from ExpertBIDecline (the decline curve module). Note that these are the parameters required for PCA.

5.4.4 Development Environment

Each programmer who is working directly in the SeTES system is identified as a SeTES developer on www.github.com, a third party code repository which is commonly used for large, multi-developer projects. The developers each have copies of the entire system installed in virtual environments (through VirtualBox or similar platforms). At the start of a development session, they download up-to-date versions of the code. After changes are made and the code is stable in their local environment, it is uploaded back to github. The full version on github is routinely installed on the Triton server at LBNL which hosts the public version of the website. Without going into a full description of github, one of its more popular features is gitblame, a utility which can be used to match code changes (particularly ones which cause bugs or a system crash) to individual developers. This keeps the development team coordinated, friendly, and on their toes.

If, eventually, a third party desires access to all of the SeTES software, they can be granted direct access through github. A disk image of the system, including the entire build, can also be provided, and this can be used to create local desktop (or laptop) copies of SeTES for private use, public demonstration, or field calculations.

Complete technical documentation for the SeTES website and systems to support it are available in Appendix I. In this section, we have described highlights of the SeTES technology only.

5.4.5 Software Ideas Developed as a Consequence of SeTES

5.4.5.1 Module Macros

One of the biggest challenges to face the SeTES development team is keeping track of all of the little details that pull the system together. The first modules took about 2 months each to develop and integrate. Now, a module can be put together in about 10 days, but the process is still tedious. Ideally, it should only take a couple of days if consistency of the underlying frameworks are maintained. Furthermore, because the modules are being put in by hand, there is considerable variation between them in terms of variable naming and input and output designations—these issues must be coordinated directly between working programmers.

To solve this problem, it was suggested by retired programmer Gary Anderson (www.Afferent.org) that we set up a system of self-compiling macros to aid in writing modules. The basic concept is of a set of module templates and a set of descriptive text templates with an interpreter sitting between them. Whenever keywords were encountered, the interpreter would substitute values taken from the text into the templates thereby essentially "writing" programs to support modules directly from readable text that would also serve as documentation. Not only would this streamline and standardize the module integration process, but it would make it much easier to communicate to the next generation of developers or third parties interested in building modified SeTES systems. Early experiments are promising.

5.4.5.2 Graphical Compiler

The second idea that was introduced in the same vein (also by Gary Anderson) is a graphical compiler algorithm, capable of rendering a flow-chart drawn in OpenOffice directly into PHP code. A modification of the fluid-selection flowchart (Figure 30) which can be compiled into PHP is shown in Figure 36.

In the compliable diagram, both questions (such as Net Pay [thickness]? and brittleness?) and conclusions (Proppant = 0, Proppant = 2) are held in the oval nodes. They are linked by possible answers to the questions. Comments are held in the square boxes. Note that the structure of this diagram is almost exactly the same as that of the Bayesian Network in section 5.2.6.1 Automatic Well Placement . The difference is that, in the compiler case, the nodes hold inputs and answers that are not probabilistic. Similar techniques can, and are, used to make implementing graphical models easier, most notable in the MCMC package BUGS.

We originally identified the need for this kind of a tool when we attempted to expand the fluid selection module to include the opinions of more than one Expert. Different petroleum engineers use different rules of thumb. We wanted the capability to sit in the office of an engineer while the "expert" draws new flowcharts. With the graphical compiler, it could be compiled into PHP of Python and displayed in real time. If a number of experts were interviewed using this process, then the opinions they produced could be compared automatically. The graphical compiler has been written but has yet to be extensively tested.

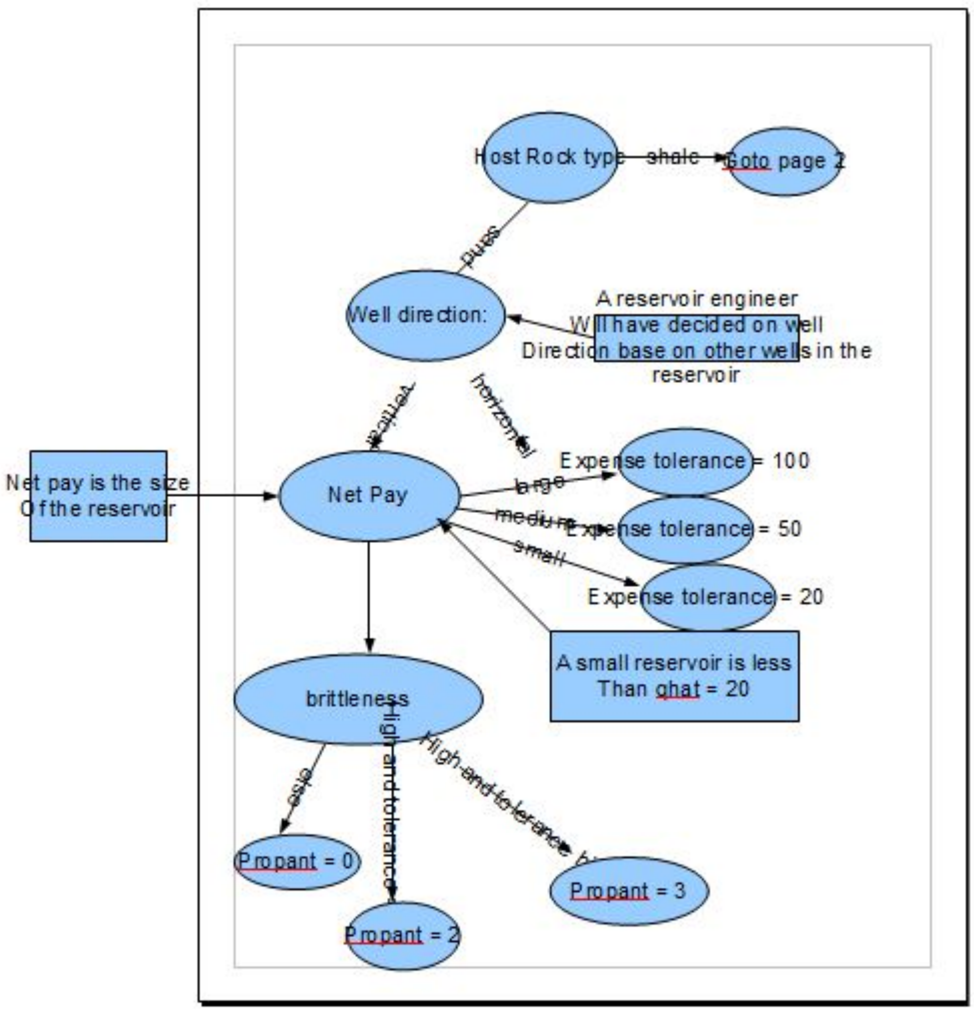

Figure 36: Fluid selection flowchart which can be compiled into PHP.

6. Results and Discussions

Despite early challenges, SeTES has met and exceeded every one of its objectives. A large part of the system is fully implemented and accessible over the internet. Further pieces of it are implemented but not currently integrated into the web experience. In addition to being the Expert System that we set out to build, it represents a new paradigm for data analysis, sharing and technical communication for the Unconventional Gas Resource community—a paradigm which can be easily adapted for other fields.

When we first started the SeTES system, we believed that we would be able to draw on off-the-shelf algorithms and programs for most of the data analysis and optimization features of SeTES. We did not want to build anything new, just bolt together existing technology to support a new application. We quickly discovered that the kind of algorithms we wanted to use had not been developed elsewhere to the level required to support unconventional resources development. For example, the first Bayesian Networks for production prediction were just beginning to be developed in 2009. Furthermore, the vast amount of data analysis in Shale Gas was being done by hand, with proprietary software, in Excel, or via Visual Basic scripts in spreadsheets, and the kind of flexible, open-source software required for high-level integration simply did not exist, nor was industry or academia accustomed to working with such tools. This meant that almost every SeTES module had to be developed from scratch (The exceptions to this rule are the PERANA shut-in analysis Fortran code which was written in the mid-1990s (but never published), the Fortran-based ShaleSIM, which was developed for high-performance computing platforms, and the Fluid Selection Menu).

Even without development of the Expert System, the SeTES project has resulted in a total of 15 publications in fundamental Shale Gas research and will probably result in as many again, as some of the newest modules are based on as-yet unpublished work. Unpublished results that have contributed to SeTES include the entire Bayesian Network algorithm, Automatic Transient Detection, Projection of parameters along seismic horizons, Automatic reading of completion reports and the new implementation of uncertainty quantification using Polynomial Chaos. This makes SeTES—even if none of it were ever deployed over the internet—a very rich project.

6.1 State of the SeTES system

Wikipedia defines the life cycle of a software project using the following terminology: (http://e.wikipedia.org/wiki/Software_release_life_cycle)

- Pre-Alpha: design and development stage and unit-testing in which individual pieces of software are tested independently

- Alpha: The first phase to begin software testing, in which software is tested internally and by a select few external tested. An Alpha-release is an internal release in which the functionality of the full system can be tested.
- Beta: All of the features of the software are complete and it can be used for demonstrations and previews and undergo rigorous user testing.
- General Availability: The point at which all of the bugs have been fixed and the software is released to the general public.

Using these definitions, SeTES has reached its Alpha phase as required and is in the early stages of Beta Development.

The SeTES alpha version released with this document contains the implemented design for a web-based Self-Teaching Expert System for the discovery and production of unconventional gas resources for which there is no direct predecessor and which has no equivalent in any field to our knowledge. In addition to satisfying all of the requirements of the project, we have:

- Implemented a prototype of the system which runs on an LBNL server and can support a small number of users
- Designed a system which is far more flexible and extensible than the original concept
- Built a complete shale-gas database framework
- Implemented a number of algorithms related to shale gas in the open programming language Python
- Conducted original research which has resulted in a large number of professional publications
- Built an entirely new platform for running Expert System modules over the web including highly sophisticated Fortran models and probabilistic inference
- Proposed a website which, in addition to presenting an expert system can be used to
    - Gather and disseminate data related to shale gas
    - Educate the public on shale gas research
    - Provide a platform for the demonstration of new methods and algorithms
    - Serve as a repository for research

6.2 Work remaining to move SeTES to a Beta deployment

In order to move SeTES to a full Beta deployment, we have yet to:

- Integrate a number of modules which have been unit-tested in the website and the full system
- Do intensive usability testing
- Integrate more than a subset of the Holly Branch Dataset including
    - Developed data uploading capacity for more than a few data formats
    - Fully test the probabilistic self-learning features of the system
- Streamline the module-integration process

As of this writing, the SeTES system is unstable for more than a few users. Parts of its structure, particularly those for self-learning have not been tested due to this lack of users. Fewer modules are incorporated than anticipated and some are not stable. Quite a few of the features of the website which are written into the code are not operational and, although the system can be run over the internet, it is a challenge (but not impossible) for a casual user to install it on an individual computer.

That said, by almost any measure, the SeTES project should be considered a success. Not only did we take an incredibly ambitious idea and concretely show how it could be accomplished, we came up with a suite of new ideas from computer learning to augment traditional petroleum engineering analysis, built an entire platform to run it over the internet, introduced a suite of new modeling tools, designed a whole new way of presenting research results, and did it in just over two years.

6.2.1 Scientific challenges

The remaining challenges to SeTES as far as its application-specific scientific side are:

- Integration of modules still in Alpha phase
- Development and implementation of new modules
- Incorporation of the complete Holly Branch Data set
    - Addition of new data - New fields and formations

A complete list of modules under development is included in the Module section.

One of the greatest scientific challenges which has not been addressed elsewhere will be, upon the integration of data from more formations and fields, to determine the similarity and differences between data and parameters from significantly different areas. It will be interesting to learn how far inferences made in one field can be extended to other fields. Although seat-of-the-pants and ballpark numbers proliferate in the UGR profession, it will be very interesting to see how well they bear up against hard numbers.

6.2.2 Technical Issues

6.2.2.1 Speed

By far the biggest limitation of the existing SeTES system is that it is too slow. Many of the high-end modules for error estimation and modeling which we would have liked to include in the Alpha version cannot be included because they take upwards of several minutes to compute on our LBNL server. Most web browsers have a set time limit that they will wait for a website to send a requested HTML file, after which they time out and print a message that the website is unavailable. Although it is possible to increase the timeout limits, it is not practical to make a casual user have to enter the advanced options menu of his browser. We developed a number of modules and features for SeTES which we are unable to deploy. These include

- Jackknife estimation of uncertainty on parameters
- Automatic transient detection
- Fracture-based modeling
- More comprehensive statistical modeling

The other issue that is posed by computational limits is that our server will support only a very few users (three or four at a time). This is fine for development, but obviously untenable for a deployed website. Many websites have computational issues, but they generally revolve around supporting thousands or millions of users at a time. Although we would like SeTES to become a popular site, we probably will never face the traffic that even small-scale commercial websites have to handle. Our issues are with a small number of users attempting extensive computation.

Very few websites offer much in the way of features that do actual computations on the server. Those that do solve the problem typically by

- Owning a huge server
- Doing computations offline and sending the user an email when results are ready
- Sending JavaScript instead of results, which punts the problem of computation by moving it onto the user’s local machine
- Working on the Cloud

The next version of SeTES will undoubtedly have to have some form of cloud implementation. Because the SeTES traffic load is uncertain, it would be impractical and prohibitively expensive to purchase a powerful server. Since results depend on each other, it does not make sense to have the user log off and then log back on later when all of the computations are completed—although this feature would be one tractable way to collect results from long-running simulations.

6.2.2.2 Simplicity and transparency

Because the current version of SeTES is the result of a research project (there could be no roadmap because the key research thrust was to determine what needed to be done), while parts of the system are very clean, including most of the PHP, other parts are best characterized as research codes—particularly the earliest modules, which were written did not make use of the best professional programming practices. Furthermore, the process of adding new modules and features to the system, which is supposed to be flexible and easy to upgrade, is actually quite tedious. Although documented, only the developers who are most closely associated with the system are currently able to add new capabilities without extensive training.

In the next iteration of SeTES, the underlying framework needs to be overhauled and simplified, in addition to being re-written for modern cloud-based deployment. Some of the ideas described in this report (templates and macros) need to be implemented so that the process of adding a new module is as simple as writing a python wrapper and filling out a form.

7. Impact on producers and others

7.1 Producers

The impact to producers of SeTES is not known at this time because the website has only been made available to a few users for usability testing. That said, much of the research underlying SeTES is now currently being made public through presentation at meetings and through journal publications (See section 8, Technology Transfer, for a current list of SeTES publications and presentations). Whether or not SeTES-sponsored research will have significant impact bottom-line impact on producers is hard to know. We have received many positive comments from producers and harsh “this will never work”-type criticisms from developers of commercial software products. But the fact is that SeTES works, and is expected to improve dramatically in the near future.

The impact of SeTES as an expert system and a platform for producing and integrating knowledge from many sources will never be realized unless users are willing to contribute data. Other than the Holly Branch data set that we used for development, and other small public data sets, we will not have enough data to fully test the self-learning features of the system unless users contribute.

Even if no data further than Holly Branch is ever added to the system, however, it is still in a position to make an important contribution to industry by providing user-friendly access to new algorithms, research and ideas not presented as technical papers at meetings, but in a hands-on functional environment. Furthermore, it is the aim of SeTES to make available basic tools for Unconventional Resource data analysis for anyone who wants to use them, including small producers, students and the public.

7.2 Academic Users

It is our hope that SeTES will become a prestigious platform for presenting research results as well as a common tool for students. Much potentially important research is lost because it only makes its way into limited forums such as professional meetings and publications. Furthermore, a common problem with research code is that often it is tuned only to one specific dataset and may or may not be stable or unbiased enough to run using other examples. SeTES provides a potentially paradigm-shifting approach to making research results concrete. In order to run over the web on user-uploaded data, an algorithm has to be bomb-proof. It is not enough to have a good idea, the idea has to be fully developed and tested. We hope that project sponsors will encourage the contribution of algorithms and software to SeTES as a form of publication.

7.3 Public

Recently, the New York Times ran an article entitled *Insiders Sound an Alarm amid a Natural Gas Rush,* (http://www.nytimes.com/2011/06/26/us/26gas.html) claiming that:

> *"In the e-mails [provided to reporters by industry insiders], energy executives, industry lawyers, state geologists and market analysts voice skepticism about lofty forecasts and question whether companies are intentionally, and even illegally, overstating the productivity of their wells and the size of their reserves."*

Although there are many websites devoted to explanations of the process, policies and environmental politics regarding shale gas in particular, almost no information is available to the public that describes, in hard terms, the kinds of analysis that go into developing and producing a field. Without delving deep in the petroleum engineering literature, there is no way for an interested, intelligent layman to educate himself on the logic behind production decisions and the uncertainty involved. SeTES is a first step in disseminating tools and information such that this important prospective energy technology can be implemented in a scientifically open, transparent fashion.

8. Technology Transfer Efforts

The SeTES website itself is the most direct form of technology transfer undertaken by this project. However, the funded work has also generated three student theses and 20 conference and journal publications. A full list of these references is given in Appendix IV.

9. Conclusions

SeTES is a self-teaching expert system that (a) can incorporate evolving databases involving any type and amount of relevant data (geological, geophysical, geomechanical, stimulation, petrophysical, reservoir, production, etc.) originating from unconventional gas reservoirs, i.e., tight sands, shale or coalbeds, (b) can continuously update its built-in 'public' database and refine the its underlying decision-making metrics and process, (c) can make recommendations about well stimulation, well location, orientation, design and operation, (d) offers predictions of the performance of proposed wells (and quantitative estimates of the corresponding uncertainty), and (e) permits the analysis of data from installed wells for parameter estimation and continuous expansion of the public data base. Thus, SeTES integrates and processes any available information from multiple and diverse sources on a continuous basis to support decision making at multiple time-scales, while expanding its internal database and explicitly addressing uncertainty.

SeTES is a web-based application that can also be downloaded and run directly on a user's facilities. It uses three types of data: public data, that have been made available by various contributors, semi-public data, which conceal some identifying aspects but are available to compute important statistics, and a user's private data, which can be protected and used for more targeted design and decision making. Semi-public dataset are afforded the level of confidentiality desired by their respective contributors. For example, the general geographical location associated with the data may be disclosed, but the data provenance may not. Thus, the user benefits from the data availability to design more productive production systems without compromising confidential information.

It is important to point out that, while significant progress has been made in the development of SeTES, significant challenges remain (as expected in an Alpha-2 release). These incluse increasing the execution speed of several modules, improving the functionality of existing modules, developing and implementing additional modules (i.e., expanded geophysics, environmental impact of stimulation are two high priorities), incorporation of new data sets into the public database, etc..

SeTES can be a vital tool in gas production from unconventional gas resources. It is expected to result in a significant increase in both reserve estimates and production by providing a technology that will increase efficiency and reduce the uncertainties associated with such gas reservoirs, thus bringing previously inaccessible energy resources to production. As a web-based application, it is also a user-friendly environment for the intuitive and concrete presentation of research results in a framework which makes it easy for a user to understand what they do and how they can be directly implemented. It is the authors' belief that SeTES introduces a new paradigm for data analysis and technical communication into the petroleum engineering community in general, not just unconventional resources.

10. Recommendations

If the SeTES project were to be dropped at this stage and never pushed into a Beta version and eventual stable release, it would be a great loss, not only for the project but for all of the people who stand to benefit from it. We would like therefore to continue onto the logical Phase II of the project, with these next-step objectives:

- Release a scaled-down website immediately
- Bring modules to completion
- Conduct rigorous user-testing
- Seek out more data
- Move to cloud-based application

FIGURES and/or TABLES

REFERENCES

Aguilera, R. 2010. “Flow Units: From Conventional to Tight Gas to Shale Gas Reservoirs.” Paper SPE presented at the Trinidad and Tobago Energy Resources Conference held in Port of Spain, Trinidad, 27-30 June 2010.

Agterberg, F.P.. 1989. “Computer Programs for Mineral Exploration.” *Science*, July 7. **245** (4913). Pp. 76-81.

Aki, K., and Richards P.G.. 1980. Quantitative seismology theory and methods. W.H. Freeman and Co.

Al-Yami, A., Schubert, J., Medina-Cetina, Z., and Yu, O.Y. 2010. “Drilling Expert System for the Optimal Design and Execution of Successful Cementing Practices.” IADC/SPE Asia Pacific Drilling Technology Conference and Exhibition, SPE 135183.

Anderson, D.M., Nobakht, M., Moghadam, S. and Mattar, L. 2010. “Analysis of Production Data from Fractured Shale Gas Well.” SPE Unconventional Gas Conference, 23-25 February. SPE 131787

Bhattacharya, S., and Nikolaou, M., 2011. “Optimal Fracture Spacing and Stimulation Design for Horizontal Wells in Unconventional Gas Reservoirs,” in SPE 147622 anticipated presentation at the Annual Technical Conference and Exhibition: Denver, Colorado, USA.

Blasingame, T.A., J.L. Johnson, W.J. Lee and R.Raghavan, 1989, “The Analysis of Gas Well Test Data Distorted by Wellbore Storage Using and Explicit Deconvolution Method,” SPE Gas Technology Symposium, Jun 7-9, SPE 19099.

Blasingame, T.A., 1996. "PERANA (Performance Analysis) -- Pressure Transient Test Analysis Software," Texas A&M University, December 1994 (last revision of the MS DOS Version).

Brillinger, D.R., 1964, “The Asymptotic behavior of Tukey’s general method of setting approximate confidence limits (the jackknife) when applied to maximum likelihood estimate.” Review of the international statistical institute, **32**. No. 3.

Brooks, R.A. 1983. "Model-based Three-dimensional Interpretations of Two-dimensional Images," IEEE Transactions on Pattern Analysis and Machine Learning. PAMI-5, 140-150.

Duda, R.O., Duda. R. O.. Harl. P. E.. Nilsson, Reboh. R., Sloeum, J, and Sutherland. G. L.. 1977. Development of a Computer-based Consultant for Mineral Exploration. Annual Report SRI Projects 5821 and 6415. I October 1976 to 30 September 1977.

Freeman, C.M., Moridis, G.J., Ilk, D., and Blasingame, T.A.: Proc. 2009 TOUGH Symposium, Berkeley, California, September 14-17 2009

Freeman, C.M., Moridis, G.J., Ilk, D., and Blasingame, T.A.: SPE 124961, Proc. SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, October 4-7 2009

Freeman, C., Ilk, D., Blasingame, T.A., and Moridis, G.J.: SPE 131583 Proc. 2010 IOGCEC International Oil & Gas Conference and Exhibition, Beijing, China, 8-10 June 2010

Freeman, C.M.: SPE Annual Technical Conference and Exhibition, Florence, Italy, September 19-22, 2010

Ghanem, R., Spanos, P. D. 1990. "Polynomial Chaos in Stochastic Finite Elements, a new method for the solution of problems: Journal of Applied Mechanics," **57**, 197-202.

Ghoraishy, S.M., Liang, J.T. and Green, D.W. 2008. "Application of Bayesian Networks for Predicting the Performance of Gel-Treated Wells in the Arbuckle Formation, Kansas." SPE/DOE Symposium on Improved Oil Recovery, 20-23. SPE113401-MS.

Ilk, D., Perego, A.D., Rushing, J.A., and Blasingame, T.A. 2008. "Exponential vs. Hyperbolic Decline in Tight Gas Sands — Understanding the Origin and Implications for Reserve Estimates Using Arps' Decline Curves." Paper SPE 116731 presented at the SPE Annual Technical Conference and Exhibition, Denver, Colorado, 21-24 September.

Holditch, S.A., Xiong, H., and Rahim, Z., 1993. "Using an Expert System To Select the Optimal Fracturing Fluid and Treatment Volume." SPE Gas Technology Symposium held in Calgary, Alberta, Canada, 28-30. SPE 26188.

Kravis, S. and Irrgang, R., 2005. "A Case Based System of Oil and Gas Well Design with Risk Assessment." Applied Intelligence 23, 39-53.

Lietard, O., Maniere J., and Norris M.: "Modeling of Transverse Hydraulic Fracturing," paper SPE 106251, Hydraulic Fracturing Technology Conference, College Station (January 29-31, 2007).

Liu, S. Raghavendra, C.S. Liu, Y.,Yao, K., Balogun,)., Olabinjo,L., Soma,R., Ivanhoe,J., Smith, B., Seren,B., Lenz, T., Dinesh Babu, C.G., Ershaghi, I. 2011. Automatic Early Fault Detection for Rod Pump Systems .
SPE Annual Technical Conference and Exhibition held in Denver, Colorado, USA, 30 October–2. SPE 146038.

McCammon, R.B. 1994. "Prospector II: Towards a Knowledge Base for Mineral Deposits." Mathematical Geology, Vol 26, 8, 916-936, DOI: 10.1007/BF02083422

Moridis, G.J. 1998. "TOUGH90: A FORTRAN90 Implementation of TOUGH2." Presented at the TOUGH98 Workshop, Berkeley, CA, May 4-6. Available at http://www.osti.gov/bridge/servlets/purl/674705-FhI0vF/webviewable/674705.pdf

Muller, J.-C., R.D. Johnson, and L.R. Vanzella. 1986. "A Knowledge-Based Approach for Developing Cartographic Expertise." Proceedings 2nd Intern'1 Symp. on Spatial Data.

Patil, A., D. Huard and C.J. Fonnesbeck. 2010. "PyMC: Bayesian Stochastic Modelling in Python." Journal of Statistical Software, 35(4), pp. 1-81.

Pfefferkorn, C., D. Burr, D 1 . Harrises, B. Heckman,, C.Oresky, and *3.* Rothermel. 1985, ACES: A Cartographic Expert System.

Pruess, K**,** Development of the general purpose simulator MULKOM, ***Rep. LBL-15500,*** Lawrence Berkeley Laboratory, Berkeley, **CA,** 1983.

Pruess, K., TOUGH2 - A general-purpose numerical simulator for multiphase fluid and heat flow, ***Rep. LBL-29400,*** Lawrence Berkeley Laboratory, Berkeley, **CA,** 199 1.

Pruess, K (ed), Proceedings of the TOUGH2 Workshop '95, ***Rep. LBL-37200,*** Lawrence Berkeley Laboratory, Berkeley, CA, 1995.

Pruess, K., A. Simmons, **Y.S.** Wu and G. Moridis, TOUGH2 software qualification, ***Rep. LBL-38383,*** Lawrence Berkeley National Laboratory, Berkeley, CA, 1996.

Reagan, M.T., Najma, H.N., Ghanemb, R.G., Knoib, O.M. 2003. Uncertainty quantification in reacting-flow simulations: Combustion and Flame **132,** 545–555.

Ramani, K. and Gautum, N.C. 1990. Expert System, for Identification of Land Use/Land Cover Categories from Satellite Data, Proceedings of the Seminar on AI, Expert System & Knowledge Based System, April 1990.

Rao, D.P. 1991. Management of Natural Resources using Expert System Preliminary Result of Some Case Study, in ***Expert system Technology in Indian Context.*** Ed. by V.V. Sharma, Deekshatulu, Vishwandhan, TMH, New Delhi.

Rao, B.R.M., Sukanaya, H., Venkatraman; Parariy A., H., and Sharda. 1990. Development of an Expert System for Classification of Soil. Proceeding of Seminar on AI, ES and KB System, ISRO, Banglore.

Rasheva, S. and Bratvold, R.B. 2011, A New and Improved Approach for Geological Dependency Evaluation for Multiple-Prospect Exploration. SPE Annual Technical Conference and Exhibition held in Denver, Colorado, USA, 30 October–2 November. SPE 147062.

Robinson, G. and Jackson, M. 1985, "Expert Systems; in Map Design," Proceesings of AUTOCARTO-7, 430-439.

Russell, S. and Norvig, P., 2003 *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall ISBN-10: 0-13-604259-7.

Saputelli, L., Nikolaou, M. & Economides, M. J. (2005). Self-learning reservoir management. *SPE Reservoir Evaluation & Engineering*, ***8***(6), 534-547.

Wu, Y.S., C.F. Ahlers, P. Fraser, A. Simmons and K. Pruess, Software qualification of selected TOUGH2 modules, Rep. LBNL-39490, Lawrence Berkeley Laboratory, Berkeley, CA, 1996.

LIST OF ACRONYMS AND ABBREVIATIONS

SeTES: Self-Teaching Expert System

EUR: Estimated Ultimate Recovery

APPENDICES

Appendix I: Development Guide

This is a short overview of some internal aspects of the SeTES application to aid developers interested in exploring and exercising the code. (Author: Ralf Santos)

1. System Overview

This document assumes one is familiar with the how to use the SeTES website as an end user and focuses on providing those who wish to develop and extend the SeTES website. This section describes the software components that comprise the SeTES site and overviews how they realize the SeTES website as users recognize it.

This document assumes a working familiarity with a variety of technologies. To orient the reader to the site's operation, some general aspects of its organizational workings are described here.

1.1 SeTES Technologies

The SeTES site uses a variety of languages and platform applications in its operation. The application languages involved are Fortran, Python and PHP. Fortran is used to implement many of the scientific codes since they are the direct product of academic research, and is ideally suited to producing performant numerically intensive code. PHP is a popular open-source language for implementing Web-based applications. Python is used to integrate the two as it is a popular open-source language that is used widely and particularly popular for systems programming as well as scripting and application integration, as it has a number of features making it easy to integrate with and embed into other applications.

The server uses a variety of support applications, all of which are open source programs:

Apache web server

Apache is the standard server used for serving web content

MySQL database

MySQL is a relational database system widely used in Web environments, and is used to provide persistent storage for information about users and files, as well as storing application state information.

Redis object store

Redis is a fast, non-persistent object store that different parts of the SeTES system use to exchange data while performing internal computations across different programs.

1.2 Site Instances

Often developing a site requires having multiple instances of that site available to do all the necessary development and testing. The SeTES application realizes this concept with system instances. Basically, the instance is a simple directory naming convention that is incorporated into code and data paths so that multiple copies of the website can be supported on the same machine. There is little explicit support for this notion (i.e. there is no central registry of instances on a machine), but it is an implicit convention that is transparent to the installed software used by developers and authors in managing the site.

The instance is a directory path component incorporated to the website file system tree and the file system paths for the expert runner code paths. So for example, two common instance names are "_setes_" and "_setes-test_". Thus the name "setes" refers to the SeTES installation comprised collectively of the website tree at `/opt/apaches/sites/setes' and the code directory `/opt/expert/setes' containing `/opt/expert/setes/python' and `/opt/expert/setes/fortran'.

The word “INSTANCE” in all capital letters will be used at various places in this document as a metasyntactic variable to refer to the name of a SeTES installation instance. So a file system path like `/opt/apache2/sites/INSTANCE/web' refers to the "web" directory of the SeTES instance named INSTANCE.

1.3 The Compute Cycle

To understand how the various portions of the site work together, it is useful to keep in mind the overall process of executing an analysis run. As this document assumes the reader is already familiar with the operation of the SeTES application, this document assumes the reader is also familiar with how the user provides input data and sets up an analysis run. Therefore, this document shall focus on detailing what happens once the user presses the button to start an analysis.

The overall compute cycle involves the following steps:

1. The web interface is implemented in a PHP-based framework called Symfony which is responsible for producing the initial page view seen by the user. The HTML of the page provides the basic structure and a JavaScript framework called jQuery supplements the basic structure with interactive features, including the launching of individual computational modules.
2. The user triggers a request to perform an analysis run via the Web interface.
3. The PHP uses a module of code called an analysis runner to gather the input data and send a JSON data structure to the Python code to execute the desired analytical tools.
4. The Python code in a program this document shall refer to as an expert runner performs the actual analytical tasks, running particular FORTRAN codes as required.
5. The Python code gathers all of the resulting data and formats it as another JSON data structure organizing the computational results.
6. The PHP code receives the data and invokes a module called an analysis renderer to unpack and format the page viewed by the user.

1.4 Code Organization

The code lives in several places:

\`/opt/apache2/sites/INSTANCE'

The website code and document tree lives here.

\`/opt/expert/INSTANCE/python'

The Python code comprising the main analysis runner lives here.

\`/opt/expert/INSTANCE/fortran'

The Fortran codes and executables are held here.

1.5 Data Organization

The data is managed both in a MySQL database and several file system trees:

`/opt/apaches/sites/INSTANCE/filestore'

This is where upload files are held.

`/opt/apaches/sites/INSTANCE/resultstore'

This is where the results of various analysis run computations are stored, including output logs and web-visible resources like image files for graphs.

2. Developer Guidelines

2.1 Developer Environments

The developer has two general options for working on SeTES code. The developer can work on the code via the website either by using a runner script or by working directly through the website proper. For working on the Python code, the developer has an additional option of working from the command line.

2.2 Website Developer Utilities

For working via the website, in addition to working directly through the website, several simple utility scripts exist:

`runner.php'

This runner is intended primarily to work on the website triton.lbl.gov

`localrun.php'

This is a variation which is available on some installations which is simply runner.php without the authentication code, for use in secure environments.

2.3 Command-Line Developer Utilities

The command-line developer utilities consist of:

1. `goose_analysis'

- This sets up input to be run by `expert_run.py'

2. `prettyjson'
   - JSON data structure pretty printer
3. `simplefiletabdump'
   - Simple dumper for the contents of the file table in the SeTES database.

Basically, doing a run consists of passing all the required environment information to the expert runner program with a JSON structure, and running the same program that the website does (/opt/expert/setes-test/python/expert_run.py). The results are returned as another JSON structure, along with a directory of files where all the generated files are held.

The script "goose_analysis.py" is used to perform the setup for an analysis run. You can use this script both to feed specific parameters as well as files to the analysis code. To feed it files, use the "-f" command line option and provide a comma-separated list of database ID's for the appropriate files in the file table in the database. For instance, "python goose_analysis -f 14,15,16" will feed information for files with ID's 14, 15 and 16 in the database. You'll need to use the Web interface to upload new files for now. To look up files already in the database, I've provided a crude file lister (`simplefiletabdump'). It doesn't take any arguments and just prints the entire table.

The script "goose_analysis.py" does two things. First, it creates a new, empty run directory for generated files, and it emits a JSON data structure appropriate to the file to standard output. To actually kick off an analysis run, the easiest way is to use a pipeline to feed the JSON output directly into the runner like this:

goose_analysis -f 14,15,16 | sudo /opt/expert/setes-test/python/expert_run.py

There are certain (mostly filesystem permission related) issues that require the "expert_run.py" script to be run as root. If you're already working as root you can drop the "sudo" in the above command.

Finally, since the JSON output can be hard to read, I've provided "prettyjson.py" which is just a simple pretty-printer for JSON data. You can do it in one line with:

goose_analysis -f 14,15,16 | sudo /opt/expert/setes-test/python/expert_run.py | prettyjson

2.4 Front End Coding

This section is still being written.

2.4.1 Front End Overview

The front end includes the PHP code which implements the user interface for the back end Python code. It provides the input features to allow the user to set up and adjust analysis runs and it gathers the output from the Python-based computation modules and formats the output. The two main products involved here are jQuery and Symfony.

2.4.2 Front End Features

From the user's perspective, there are three primary objects of interest:

1. **Data Manager**
   - Used to search for available files, upload new ones, and modify reservoir properties.
2. **Selected Files Pane**
   - Shows the details of files and lets user select the precise set of files to use in the next analysis run.
3. **Results Pane**
   - Lets user navigate among the various computational results.

2.4.3 Front End Tab Mechanics

The navigational tabs at the top of a pane are implemented as a list which is rendered horizontally. The list items use an attribute `_content' to identify a sub-pane which corresponds to the content to display for that tab.

The front end uses a tab navigation convention to organize content. This convention is intimately connected with how the front end communicates with the back end. To illustrate this convention we'll examine a portion of the analysis pane which presents the most complete example of this convention:

```
<div class="analysis-results r_results" style="display: none;">

  <h3>Analysis Results</h3>

  <ul class="tabs clearfix">

    <li class="selected" _content="tab-production-analysis">

     <a class="analysis-tab" href="#">Production Analysis</a></li>
```

```
        <li _content="tab-geology">
          <a class="analysis-tab" href="#">Geology</a></li>
        <li _content="tab-simulation-optimization">
          <a class="analysis-tab" href="#">Simulation/Optimization</a></li>
        <li _content="tab-well-design">
          <a class="analysis-tab" href="#">Well Design</a></li>
      </ul>
      <?php include_partial('analysis_tab_ProductionAnalysis',
                 array('show' => true)); ?>
      <?php include_partial('analysis_tab_Geology'); ?>
      <?php include_partial('analysis_tab_SimulationOptimization'); ?>
      <?php include_partial('analysis_tab_WellDesign'); ?>
    </div>
```

When one looks inside the `_analysis_tab_ProductionAnalysis.php' file, one sees the following:

```
    <div class="analysis-tab tab-production-analysis"
      <%= isset($show) && $show ? '' : ' style="display:none;"' %>
    >
      <div class="button-nav">
        <button type="button" class="small analysis-button selected"
            _group="pa" _content="result-panel-BIDecline">
          Decline Curve
        </button>
      <!-- [... material deleted ...] -->
      <div class="result-panel result-panel-pa result-panel-BIDecline"
```

```
        _expertCall="BIDecline"></div>

      <!-- [... material deleted ...] -->

   </div>
```

So the analysis pane implements a second layer of sub-pane navigation via the button elements in each tab sub-pane, but fundamentally the mechanics of the process are the same. The `_content' attribute on the item clicked directs the code to the sub-pane to display for that selection. In the case of this last layer, the content isn't available in a template, but is produced by the analysis renderer after the analysis runner dispatches the appropriate computation. This process is directed by the `_expertCall' attribute. This will be discussed in the next section.

2.4.4 SeTES jQuery Plugin Overview

The JavaScript code used by SeTES is organized as a jQuery plugin. In the main, the general structure of the code should be familiar to those who are familiar with writing jQuery extensions. Overall, the plugin is comprised of exactly two statements:

- Assign to the Javascript global variable `setes.workflow' which takes a single argument and defines all of the methods and event binders used by seTES as extensions of that object.
- Run the `setes.workflow' object on `jQuery' when the document is loaded.

The code defines all of its data within function closures to minimize contamination from external code and avoid namespace pollution. Thus everything it uses either comes from jQuery or within the function `setes.workflow'. As all variables defined in the body of the `setes.workflow' function are not visible externally or bound to any external namespace, they can only be seen from within the function.

The motif of using function closures to extend passed-in objects is extended within the plugin to implement the user interface extensions, though this motif might be seen most easily if one reads the `setes.workflow' code backwards from the end to the beginning rather than in the normal order. A master private function `_.initialize' performs a series of modifications to the current HTML page by looking up the DOM nodes for the major user interface components recognized by the user (the Data Manager, the Selected Files panel, the Results panel) and invoking initializer functions written to implement the appropriate behaviors on each of them.

Several key private methods within `setes.workflow' most directly involved in the computational cycle include:

`_.createExpertCallQueryData'

> This method actually collects the set of selected files from the file manager and all parameter data from input controls on the page and packages them in a JavaScript object to be written into JSON format for the analysis runner.

`_.makeExpertCall'

> This is the central function which performs the Ajax call to the Python expert runner code.

`_.processAnalysisResultResponse'

> This performs all the client-side work of rendering a set of analysis results. It takes the data values provided by the PHP analysis renderer module and updates the appropriate elements of the HTML page, along with several housekeeping functions like marking which page elements are loaded and updating controls like selector elements used to page between different well results.

2.4.5 PHP and Symfony Coding

2.4.5.1 Overview of PHP and Symfony

As hinted at earlier, much of the front end functionality is organized and managed by a PHP-based Web application framework called Symfony (http://www.symfonyproject.org). Symfony implements an MVC (Model-View-Controller) framework on top of PHP, which in rough terms splits the application functionality into a component which manages the underlying data and state information (the Model), a component focused on managing the display and presentation of that data (the View) and a component which coordinates the first two to implement logic and behavior (the Controller).

All of the SeTES front end is implemented as a single application creatively called `frontend' which conventionally lives in the directory `/opt/apache2/sites/INSTANCE/apps/frontend'. Within the `frontend' application one will find under the subdirectory "modules" three modules:

1. `workflow'

2. `index' (implements the entry page and performs access control, validating logins)
3. `error' (performs error handling)

2.4.5.2 Front End Template Development

2.4.5.3 Front End Tab Development

To develop a new front end tab, one must begin by adding a list element and adding an HTML `div' element for your content. The important thing is that the `div' element includes a class name that corresponds to the `_content' attribute in your list element. Also, the `_expertCall' attribute must name the appropriate Python module to run from the expert runner. This should create the pane for your new module and launch the appropriate code.

To complete the process, one must provide for rendering the returned results by defining an analysis renderer module. This is described in a later section.

2.4.5.4 Analysis Runner Coding

Runner coding is only required if one wants to substantially alter how analysis runs are performed or how the analysis process is launched. For most purposes, it is adequate to use the pre-existing analysis runner and build renderers appropriate to your task.

2.4.5.5 Analysis Renderer Coding

To present the results of a module one must code an analysis renderer and a companion template. The template should contain the content of the page with variable values representing the bits of content from the computational results provided from the analysis run. The template must be paired with an analysis renderer class that knows how to unpack the computational results and arrange them to match the variable values required by the template.

2.5 Back End Coding

Back end coding involves working with the Python scripts in `/opt/expert/*/python'.

2.5.1 SeTES Python Overview

The `expert_run.py' script is the main entry point for the Python but does little on its own aside from gathering the input and delivering the output.

Important Python classes include:

- `Application'
  - In contrast to other frameworks where the "Application" refers to a top-level object, this is the class for individual modules.
- `Session'
  - This is the class for the Web session.
- `UserSession'
  - This actually refers to a component of the Session object, one which is passed between, using the Redis object store to pass data between computational modules.

2.5.2 Module Conventions

The main features of defining a module are defined by the class `ExpertBase'. Most are implemented as methods, but some are simply conventional practices.

Important class variables:

`_requiredInputs'

List of variable names needed by module.

`_dependsOn'

Lists module(s) required to obtain variables.

`run'

Defined by the module author. Main function performing core module function.

`bootstrap'

Defined by the module author when special steps are required to prepare data prior to module execution.

`extractRequiredInputs'

Pre-defined method acquired by inheritance from `ExpertBase' class. Used by module author to obtain input variables declared via `_requiredInputs'.

2.5.3 Module Input Dependencies

Getting module input involves declaring what variables are required and any prerequisite modules required to compute, then actually retrieving the appropriate variables. Getting inputs involves setting several class variables which are referenced when a module is run. They are:

`_requiredInputs'

> An array of variable names. The named variables are retrieved from the user session when the method `self.extractRequiredInputs()' is invoked.

`_dependsOn'

> An array whose sole item is a module name.

2.5.4 Module Outputs

Several functions are used to post results produced by a module:

`self._session.addOutputParam'

> Takes a variable name and the variable value as arguments. Primary mechanism for posting parameters.

`self._session.addWellResults'

> Takes two arguments: a well ID and a Python dictionary of variable names and values as arguments. Used to post results representing properties specific to particular wells.

Appendix II: Decline Curve Analysis Interium Report

Decline Curve Analysis and Production Analysis for Holly Branch Field Kernel Wells

Our methodology for the Holly Branch Field kernel wells is twofold. First we perform decline curve analysis for the kernel wells. We utilize the empirical "power-law exponential" rate decline relation (Ilk, *et al*. 2009) for decline curve analysis. The "power-law exponential" rate decline relation is given as:

$$q_g(t) = \hat{q}_{gi} \exp[-D_\infty t - \hat{D}_i t^n] \quad \text{.......(1)}$$

Our objective is to estimate the recovery (*EUR*) (or extrapolated cumulative gas production) at a specified time limit using Eq. 1 (in our case, we chose 30 years). The analysis results are presented in **Table 1** and the *EUR* map based on the kernel wells of the Holly Branch Field is given in **Fig. 1**. Our second goal is to establish a correlation between the power-law exponential model parameters and the well/reservoir properties such as permeability (k), fracture half-length (x_f), contacted gas-in-place (*CGIP*), *etc*. To that end, we perform model-based analysis using an analytical solution for a vertical well with a hydraulic fracture — see Pratikno, *et al*. (2003) and Ilk, *et al*. (2007) for orientation on production analysis for hydraulically fractured wells.

Table 2 presents the results of the model-based production analysis for the kernel wells in Holly Branch Field. In **Fig. 2** we present a map of permeability obtained from production analysis. We note that the permeability ranges between 0.0085 md and 0.04 md with the exception of one well (0.07 md). In all of our analyses we use the "finite conductivity" fracture model and our results confirm low to medium fracture conductivities. To model the drainage area, we use a rectangular reservoir configuration, and accordingly, the drainage aspect ratio (ξ) is defined as the ratio of the length (l) (East-West direction) to the width (w) (North-South direction) of the rectangle. Based on the drainage area, aspect ratio and fracture half-length we define the dimensionless "characteristic length," which is the half of the diagonal of the rectangle and given as:

$$l_{ch} = \frac{2x_f}{w\sqrt{1+\xi^2}} \qquad (\xi = l/w) \quad \text{.......(2)}$$

The characteristic length correlates the fracture half-length with drainage area configuration. We present the map of the characteristic length in **Fig. 3**.

As mentioned earlier, our primary objective is to correlate the power-law exponential rate parameters with the model-based production analysis results. We first attempted to obtain correlations by cross-plotting a variety of combinations of the power-law exponential rate model parameters versus the fractured well model variables. These cross-plots suggest that three correlations can be obtained from our results. First, the permeability (k) can be correlated with the power-law exponential rate model parameters; this correla-tion is given as:

$$k = a\, n^b\, \hat{D}_i^{\,c}\, \hat{q}_{gi}^{\,d} \quad \text{.......(3a)}$$

$$k = 0.03224\, n^{1.18341}\, \hat{D}_i^{\,-0.1052}\, \hat{q}_{gi}^{\,0.1775} \quad \text{.......(3b)}$$

In **Fig. 4** we present the calculated permeability values versus the permeability values obtained from our "permeability" correlation.

Next, we develop the "permeability -fracture half-length" (k-x_f) correlation. We correlate the permeability-fracture half-length product versus the n-parameter in power-law exponential rate decline model. This correlation is given as:

$$k\, x_f = \alpha \exp[\beta n] \quad \text{.......(4a)}$$

$$k\, x_f = 2.068 \exp[6.780 n] \quad \text{.......(4b)}$$

The comparison of the calculated permeability-fracture half-length values versus the production analysis fracture half-length results is given in **Fig. 5**.

Finally we correlate the characteristic length versus the estimated ultimate recovery (*EUR*). **Fig. 6** shows that a "linear" correlation can be obtained, and this correlation is given as:

$$l_{ch} = m\,EUR + n \quad \text{(5a)}$$

$$l_{ch} = 0.1029\,EUR + 0.0292 \quad \text{(5b)}$$

In conclusion the correlations developed in this work indicate that the well/reservoir properties can be estimated using the parameters from the power-law exponential relation (Eq. 1). The correlations developed for the kernel wells can be used in a prediction mode for obtaining the well/reservoir properties of the remaining wells in the Holly Branch Field.

References:

Ilk, D., Rushing, J.A., Sullivan, R.B., and Blasingame, T.A. 2007. Evaluating the Impact of Waterfrac Technologies on Gas Recovery Efficiency: Case Studies Using Elliptical Flow Production Data Analysis. Paper SPE 110787 presented at the SPE Annual Technical Conference and Exhibition, Anaheim, California, 11-14 November.

Ilk, D., Perego, A.D., Rushing, J.A., and Blasingame, T.A. 2008. Exponential vs. Hyperbolic Decline in Tight Gas Sands — Understanding the Origin and Implications for Reserve Estimates Using Arps' Decline Curves. Paper SPE 116731 presented at the SPE Annual Technical Conference and Exhibition, Denver, Colorado, 21-24 September.

Pratikno, H., Rushing, J.A., and Blasingame, T.A. 2003. Decline Curve Analysis Using Type Curves — Fractured Wells. Paper SPE 84287 presented at the SPE Annual Technical Conference and Exhibition, Denver, Colorado, 5-8 October.

Table 1: Holly Branch Field decline curve analysis

DCA results for the kernel wells (power law exponential rate decline model is used for decline curve analysis)

Well Name,	$\hat{q}_{gi}$ (MSCFD)	$\hat{D}_i$ (D^{-1})	n (dim.less)	D_∞ (D^{-1})	EUR (BSCF)
Abe-Jones A1	23,465	1.01	0.17	1.00E-09	4.75
Dynegy C1	3,448	0.31	0.28	1.00E-09	2.17
Knowles A1	30,309	1.69	0.12	5.03E-05	3.53
Knowles A2	9,494	0.66	0.20	1.00E-09	3.67
Knowles A3	5,626	0.63	0.20	1.00E-09	2.79
Perrenot A1	4,483,360	5.00	0.05	1.00E-09	2.16
Pickens B1	631,738	5.00	0.06	1.00E-09	2.43
Reed C2	39,722	2.01	0.12	1.00E-09	2.62
Williford C3	2,458	0.83	0.18	1.90E-04	0.45
Cunningham A3	28,326	0.37	0.33	1.00E-09	3.88
Dynegy C2	1,493,669	5.00	0.07	1.00E-09	3.54
Louetta Parker A2	67,518	2.05	0.12	1.00E-09	3.89

Table 2: Holly Branch Field production analysis (PA) results for the kernel wells

Well Name	k (md)	x_f (ft)	k-x_f (md-ft)	F_{cD} (dim.less)	Asp.Ratio (dim.less)	$CGIP$ (BSCF)	l_{ch} (dim.less)
Abe-Jones A1	0.0170	370	6.290	5.0	3	3.870	0.4680
Dynegy C1	0.0400	350	14.000	1.5	3	2.510	0.3074
Knowles A1	0.0300	170	5.100	20.0	6	4.590	0.1118
Knowles A2	0.0200	350	7.000	0.5	6	4.620	0.3335
Knowles A3	0.0175	450	7.875	0.8	4	3.300	0.4198
Perrenot A1	0.0085	250	2.125	8.8	4	1.740	0.2526
Pickens B1	0.0115	367	4.221	7.1	4	2.870	0.3123
Reed C2	0.0057	270	1.539	6.5	3	2.900	0.4269
Williford C3	0.0280	160	4.480	1.1	3	0.512	0.2530
Cunningham A3	0.0720	400	28.800	5.6	3	4.310	0.5060
Dynegy C2	0.0270	334	9.018	23.0	3	2.320	0.4400
Louetta Parker A2	0.0159	338	5.374	1.3	4	3.480	0.4822

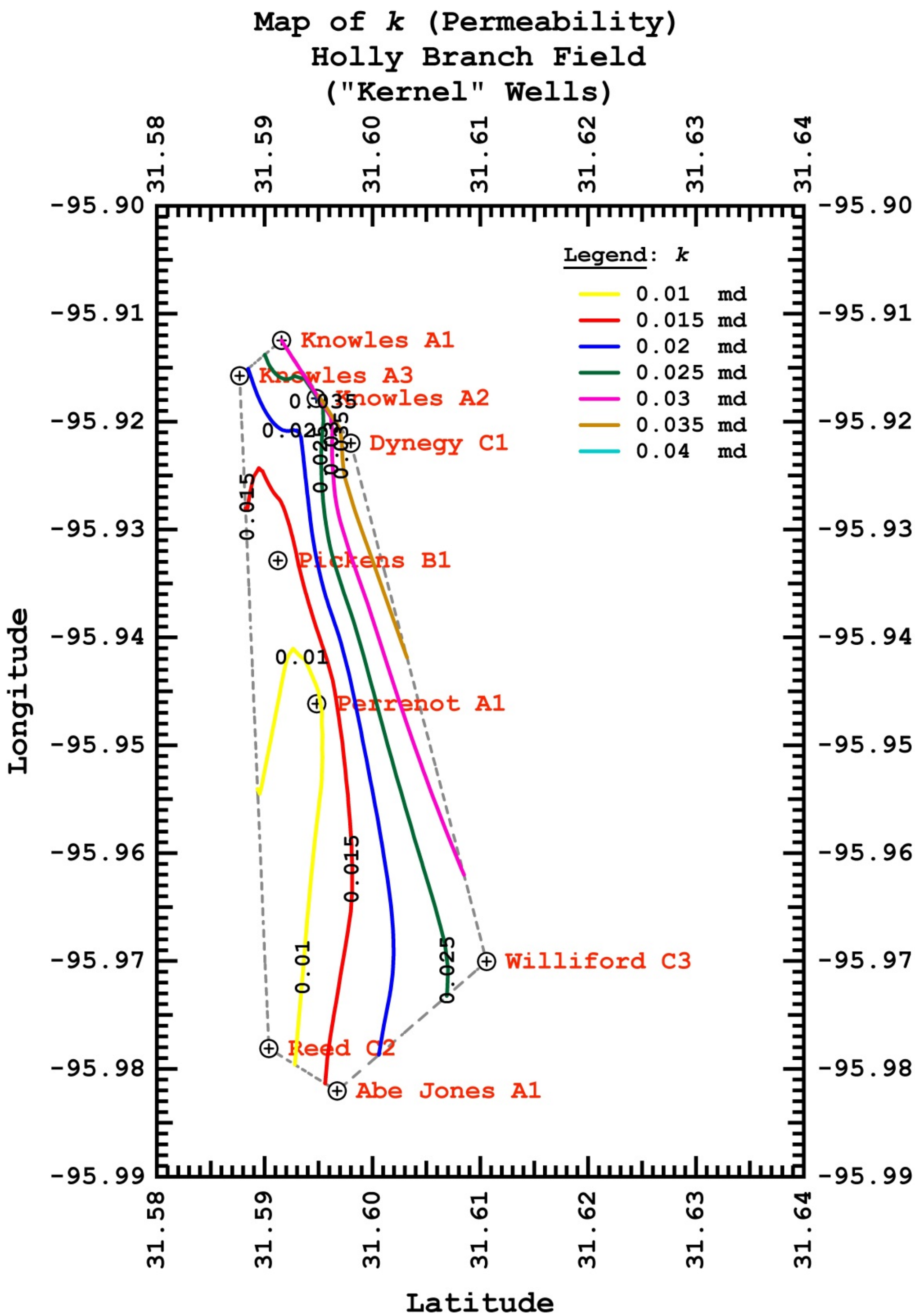

Figure 01: Map of *k* (Permeability) — Holly Branch Field ("Kernel" Wells)

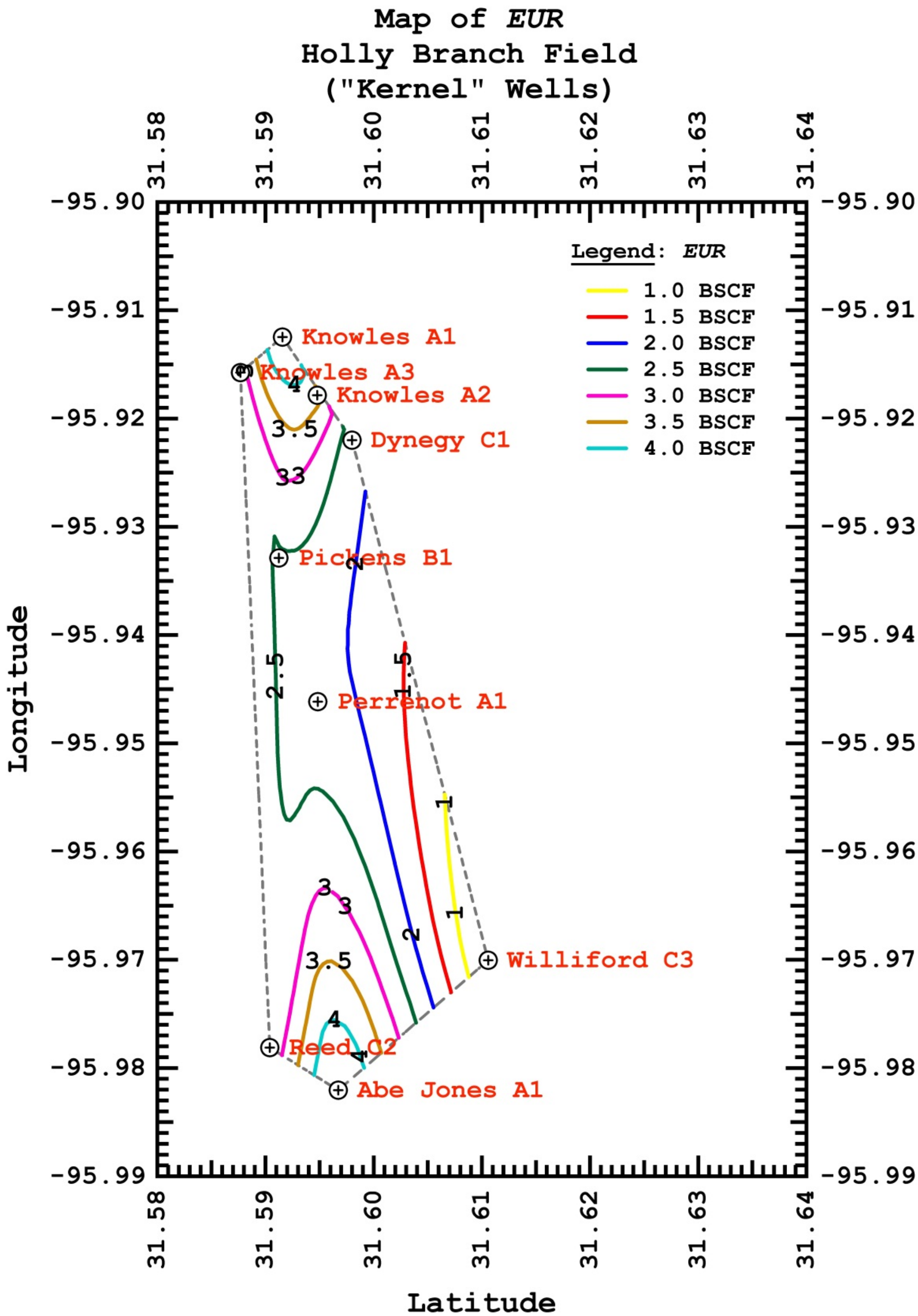

Figure 02: Map of *EUR* (Estimated Ultimate Recovery) — Holly Branch Field ("Kernel" Wells)

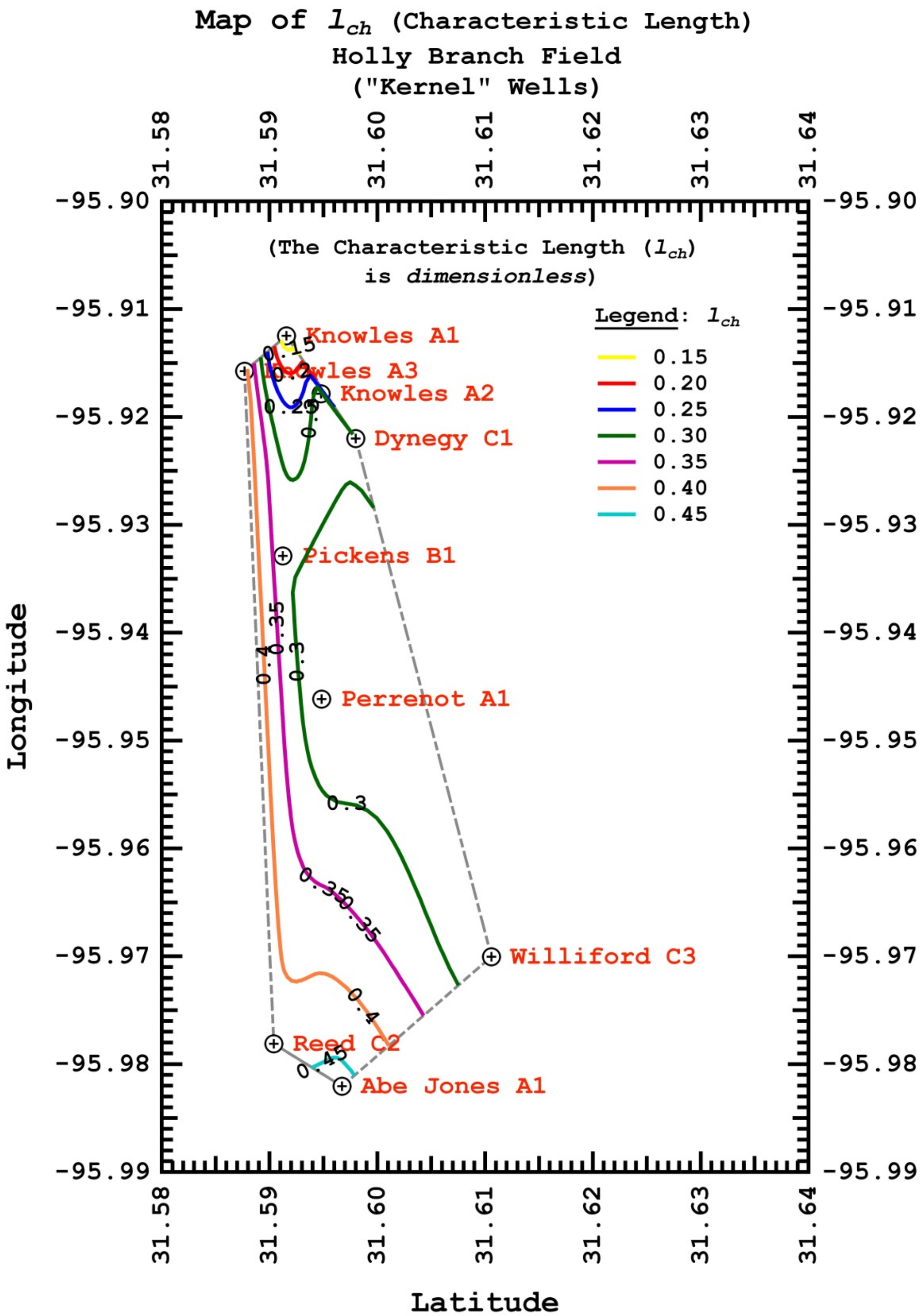

Figure 03: Map of l_{ch} (Characteristic Length) — Holly Branch Field ("Kernel" Wells)

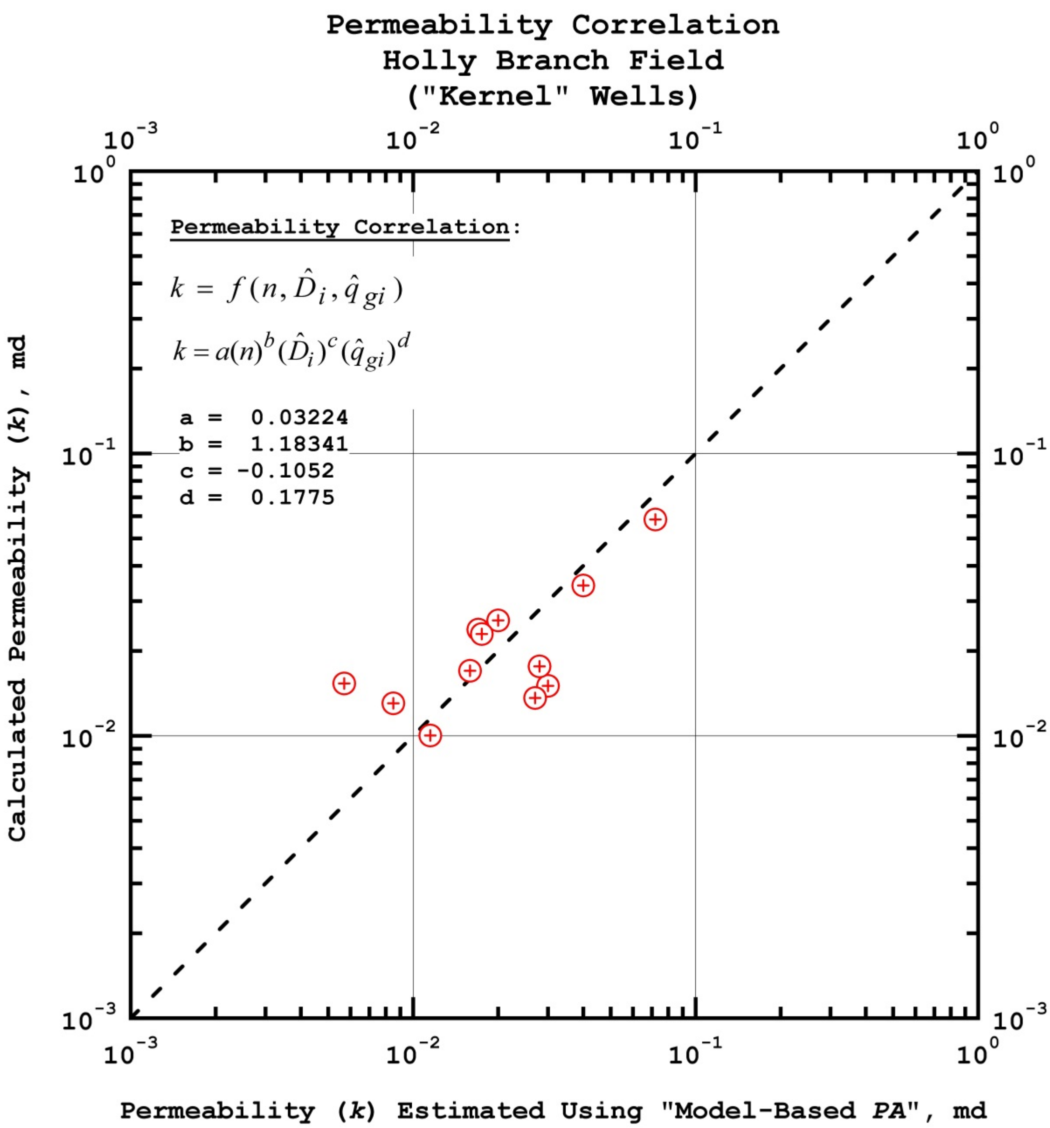

Figure 04: Comparison of permeability

Calculated using the permeability correlation versus the permeability obtained using "model-based" production analysis — Holly Branch Field ("Kernel" Wells)

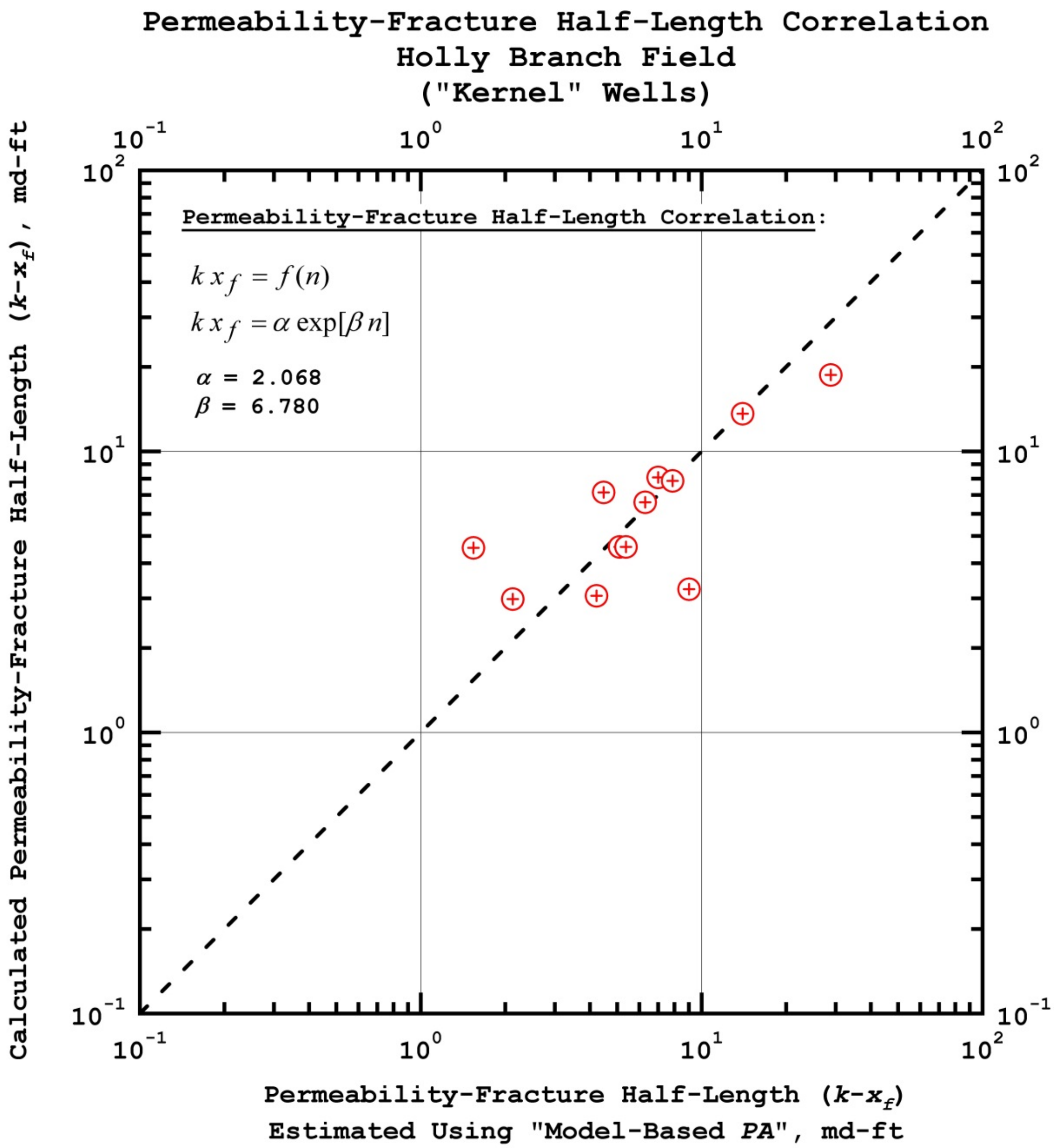

Figure 05: Comparison of permeability-fracture half-length

Calculated using the permeability-fracture half-length correlation versus the permeability-fracture half-length obtained using "model-based" production analysis — Holly Branch Field ("Kernel" Wells)

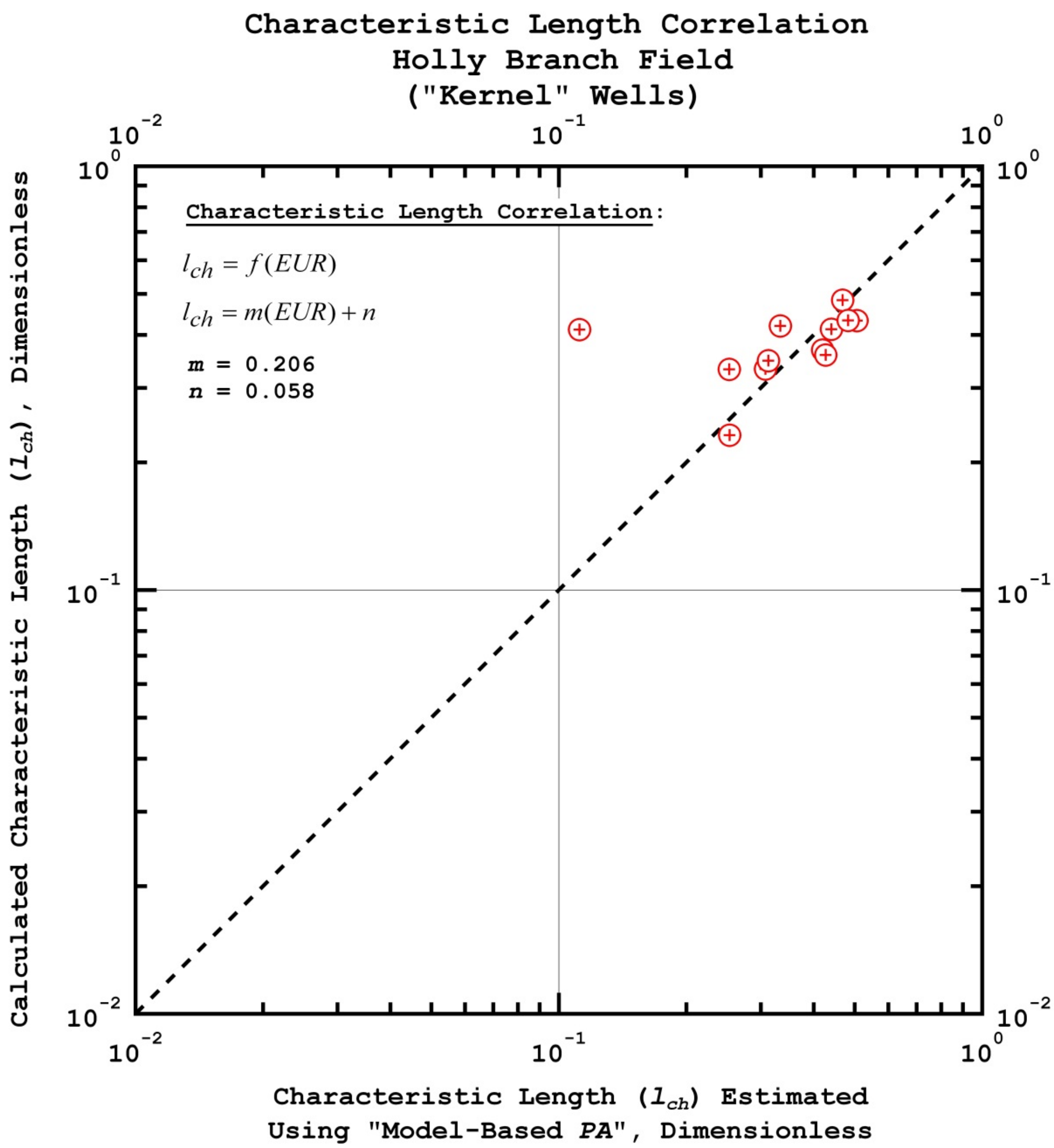

Figure 06: Comparison of Characteristic Length

Calculated using the characteristic length correlation versus the characteristic length obtained using "model-based" production analysis — Holly Branch Field ("Kernel" Wells)

Appendix III: List of Comparable Websites

There are no websites comparable to SeTES in terms of offering a set of analysis and prediction tools at no cost to the user; access to all of the math and programming behind those tools; access to current research related to shale gas; and access to data, all in a user-friendly environment. However, there are other companies with comparable software tools and support available for a fee and organizations with useful information on shale gas. There are several other RPSEA-funded projects with deliverables available through the RPSEA website (and we expect more to appear as projects wrap up).

All of these links will be made available on the SeTES website through a “Links” tab. This links listing is intended to be suggestive and not exhaustive, at least initially. We will try to include all the major players, the promising smaller players, and any relevant and interesting projects we can find through web search or recommendation. The list will be updated regularly.

Organizations and entities related to shale gas (including RPSEA projects):

Web Address	Offerings	Location/Affiliation	Notes
rpsea.com	Research funding, newsletters, events	TX producer group	
pagaslease.com	Info/forum for PA landowners/leaseholders	Pennsylvania/Marcellus	useful links page
shaleblog.com	News about oil and natural gas from shale	North America	mostly links to articles; not original content
peer.caltech.edu or peeri.org	Petroleum Energy & Enviro. Res. Center	Caltech	see geoisochem above
rtec-rtp.org	Research Triangle Energy Consortium	Duke, NC State, UNC, RTI Int'l	Univ consortium
fracfocus.com	Chemical Disclosure Registry	Groundwater Protection Council	
efdsytmes.org	A risk-based decision optimization tool	TAMU, HARC	stochasticgeomechanics.civil.tamu.edu/efd/
geology.com/oil-and-gas/	Articles and basic information, maps	HQ Pennsylvania	run by retired geologist/professor
energytomorrow.org	Information/promotion	API	promoted in recent television commercials
energynation.org	Information/promotion	API	petro industry workers
energycitizens.org	Information/promotion	API	citizens' group
aqwatec.mines.edu/produced_water/	Produced Water Treatment and Beneficial Use Information Center	CSM & Argonne	RPSEA, includes analysis module tools
geology.utah.gov/emp/shalegas/paleo_shalegas/index.htm	Research and Well Data	Univ of Utah, Utah Geologic Survey	RPSEA
geology.utah.gov/emp/tightgas	Study integrating natural and hydro fracking	Univ of Utah, Utah Geologic Survey	RPSEA

Web Address	Proprietary Software	HQ Location	Notes
slb.com	Petrel, others	Houston	
ocean.slb.com	Ocean	Houston	open-application framework
halliburton.com	Max3Di, others, partnered w/LMKR?	Houston	owns Sperry Drilling and Landmark Graphics
bakerhughes.com	JewelSuite	Houston	also jewelsuite.com
enersight.com	WellSpring	Calgary CANADA	US office in Houston
transformsw.com	TerraView, TerraFusion, others	Houston, CO, CANADA	
caesarsystems.com	PetroVR	Houston	
lmkr.com	GeoGraphix, Object Reservoir, Envertor	Calgary CANADA	alliance w/objectreservoir and Landmark/Halliburton
objectreservoir.com	ORKA: Limits, Resolve, Saige, Orpheus	Houston	see LMKR
kappaeng.com	Ecrin	France, UK, Houston	
earthsoft.com	EQuIS	Concord, MA	environmental data management w/ArcGIS
esri.com	GIS, ArcGIS	Redlands, CA	mapping, not spec to petro engr.
fekete.com	Fekete Harmony, F.A.S.T. series	Calgary CANADA	
petex.com	IPM Suite	UK, Houston	
tgsnopec.com		global, Houston	data library of seismic imaging, well data and interpretive products and services
leaderdrilling.com	DrillRight	UK	
techdrill.com	Techdrill Software Platform, DSP-One	UK, Houston	drilling software
engineeringcomputation.com	Neotec WELLFLO	no address on website	
senergyworld.com	Interactive Petrophysics, Oilfield Data Mgr	UK	global energy services company
upstreamprofessionals.com	UPI Total Asset Manager (TAM) Solution	Houston	
makinhole.com	SES	Golden, CO	note website name
geoisochem.com	GOR-Isotopes	Covina, CA	CalTech guys, Industrial Associate Research and Development Program funding

Appendix IV: List of Publications from SeTES Project

Research Papers

Currie, S.M., Ilk, D., and Blasingame, T.A.: *Continuous Estimation of Ultimate Recovery*. SPE 132352, Proc. SPE Unconventional Gas Conference, Pittsburgh, Pennsylvania, 23-25 February 2010: http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-132352-MS&soc=SPE

Currie, S.M., Ilk, D., Blasingame, T.A., and Symmons, D.: *Application of the "Continuous Estimation of Ultimate Recovery" Methodology to Estimate Reserves in Unconventional Reservoirs*, paper CSUG/SPE 138155 was presented at the Canadian Unconventional Resources & International Petroleum Conference held in Calgary, Alberta, Canada, 19-21 October 2010: http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-138155-MS&soc=SPE

Freeman, C.M., Moridis, G.J., Ilk, D., and Blasingame, T.A.: *A Numerical Study of Microscale Flow Behavior in Tight Gas and Shale Gas Reservoir Systems*. Proc. 2009 TOUGH Symposium, Berkeley, California, September 14-17 2009: http://www.rpsea.org/attachments/contentmanagers/3443/07122-23_TOUGH_2009_Symposium_Presentation_9-14-09.pdf

Freeman, C.M., Moridis, G.J., Ilk, D., and Blasingame, T.A.: *A Numerical Study of Performance for Tight Gas and Shale Gas Reservoir Systems*. SPE 124961, Proc. SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, October 4-7 2009: http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-124961-MS&soc=SPE

Freeman, C., Ilk, D., Blasingame, T.A., and Moridis, G.J.: *A Numerical Study of Tight Gas/Shale Gas Reservoirs - Effects of Transport and Storage Mechanisms on Well Performance*, SPE 131583 Proc. 2010 IOGCEC International Oil & Gas Conference and Exhibition, Beijing, China, 8-10 June 2010: http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-131583-MS&soc=SPE

Freeman, C.M.: *A Numerical Study of Microscale Flow Behavior in Tight Gas and Shale Gas Reservoir Systems*. SPE Annual Technical Conference and Exhibition, Florence, Italy, September 19-22, 2010: http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-141125-STU&soc=SPE

Freeman, C.M., Moridis, G.J., and Blasingame, T.A., 2010*. "A Numerical Study of Microscale Flow Behavior in Tight Gas and Shale Gas Reservoir Systems"*. Transport in Porous Media (In Review): http://www.springerlink.com/content/y814156x25q4wv3x/

Ilk, D., Rushing, J.A., and Blasingame, T.A.: *"Decline-Curve Analysis for HP/HT Gas Wells: Theory and Applications"* SPE 125031 Proc. 2009 SPE Annual Technical Conference and Exhibition, New Orleans, LA, USA, 4-7 October 2009: http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-125031-MS&soc=SPE

Ilk, D. Currie, S.M., Symmons, D., Rushing, J.A., and Blasingame, T.A.: *"Application of the "Power Law Hyperbolic" Rate-Decline Model for the Analysis of Production Performance in Unconventional Reservoirs,"* SPE 135616 Proc. 2010 SPE Annual Technical Conference and Exhibition, Florence, Italy, 19-22 September 2010: http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-135616-MS&soc=SPE

Ilk, D., Currie, S.M., Rushing, J.A., and Blasingame, T.A.: *"Production Analysis and Well Performance Forecasting of Tight Gas and Shale Gas Wells,"* SPE 139118 Proc. 2010 SPE Eastern Regional Meeting, Morgantown, WV (USA), 12-14 October 2010: http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-139118-MS&soc=SPE

Ilk, D., Rushing, J.A., and Blasingame, T.A.: *"Integration of Production Analysis and Rate-time Analysis via Parametric Correlations - Theoretical Considerations and Practical Applications,"* SPE 140556 Proc. SPE Hydraulic Fracturing Technology Conference and Exhibition, The Woodlands, Texas, USA, 24-26 January 2011: http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-140556-MS&soc=SPE

Johnson, N.L., Currie, S.M., Ilk, D., Blasingame, T.A.: "A Simple Methodology for Direct Estimation of Gas-in-place and Reserves Using Rate-Time Data," SPE 123298 Proc. 2009 SPE Rocky Mountain Petroleum Technology Conference, Denver, CO, 14-16 April 2009: http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-123298-MS&soc=SPE

Kuzma, H. A., Zhao, Y., Reagan, M.T., Rector, J.W.: "Polynomial chaos for uncertainty quantification in geophysics," SEG Expanded Abstracts 30, 2742-2746 (2011): http://library.seg.org/vsearch/servlet/VerityServlet?KEY=SEGEAB&smode=strresults&sort=rel&maxdisp=25&threshold=0&pjournals=SEGLIB%2CGPYSA7%2CLEEDFF%2CJEEGXX%2CSEGEAB%2CGMALCH%2CSAGEEP%2CSEGBKS&possible1=kuzma&possible1zone=article&OUTLOG=NO&viewabs=SEGEAB&key=DISPLAY&docID=3&page=0&chapter=0

Mattar, L., Gault, B., Morad, K., Clarkson, C., Ilk, D., Blasingame, T.A.: *"Production Analysis and Forecasting of Shale Gas Reservoirs: Case History-based Approach,"* SPE 119897 Proc. 2008 Shale Gas Production Conference, Irving, TX, USA, 16-17 November 2008: http://www.pe.tamu.edu/blasingame/data/0_TAB_Public/TAB_Publications/SPE_119897_(Mattar)_Prd_Anl_y_Focst%20_Shale_Gas_Hist_Based_Ap_(wPres).pdf

Moridis, G.J. and Blasingame, T.A.: *"Analysis of Mechanisms of Flow in Fractured Tight Gas Reservoirs"*. SPE 131644 Proc. SPE International Oil & Gas Conference and Exhibition held in Beijing, China, June 8-10, 2010: http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-139250-MS&soc=SPE

Nazari, S., Kuzma, H.A., Rector, J.W.: "*Predicting permeability from well log data and core measurements using support vector machines*," SEG Expanded Abstracts 30, 2004-2008, 2011. http://library.seg.org/getpdf/servlet/GetPDFServlet?filetype=pdf&id=SEGEAB000030000001002004000001&idtype=cvips&prog=search&bypassSSO=1

Theses

Freeman, C.M. 2010. Study of Flow Regimes in Multiply-Fractured Horizontal Wells in Tight Gas and Shale Gas Reservoir Systems, MS Thesis, Texas A&M U. College Station, TX: http://www.pe.tamu.edu/blasingame/data/0_TAB_Grad/TAB_Grad_Thesis_Archive/MS_043_FREEMAN,Matt_Thesis_TAMU_(May_2010).pdf

Currie, S.M. 2010. Application of the Continuous EUR Method to Estimate Reserves in Unconventional Gas Reservoirs. MS thesis, Texas A&M University, College Station, TX: http://www.pe.tamu.edu/blasingame/data/z_web_Archive/100621_SMC/100622_Thesis_Body_Chps_1_2_3.pdf

Olorode, O.M. 2011. Numerical Modeling Of Fractured Shale-Gas And Tight-Gas Reservoirs Using Unstructured Grids, MS Thesis, Texas A&M University, College Station, TX.